inzynierka

December 7, 2021

1 cloud setting

[2]:

```
BUCKET_NAME = "praca_inzynierska_image_recognition"
REGION = "us-central1"

! gsutil ls -al gs://$BUCKET_NAME
! sudo gsutil cp gs://$BUCKET_NAME/$VIDEO_NAME /home/jupyter/resources/
 ↪$VIDEO_NAME
```

```
 104191410  2021-10-23T17:19:41Z
gs://praca_inzynierska_image_recognition/video.mp4#1635009581502033
metageneration=1
                                 gs://praca_inzynierska_image_recognition/keras-
job-dir/
TOTAL: 1 objects, 104191410 bytes (99.36 MiB)
InvalidUrlError: Cloud URL scheme should be followed by colon and two slashes:
"://". Found: "gs:///".
```

2 video edit

2.0.1 setup video var

[3]:

```
VIDEO_NAME = "video.mp4"
VIDEO_PATH = f"/home/jupyter/resources/{VIDEO_NAME}"

VIDEO_OUTPUT = f"/home/jupyter/resources/out/vid2.mp4"
IMAGE_OUTPUT = f"/home/jupyter/resources/out/pngs/image-%07d.png"

IMAGES_PATH = "/home/jupyter/resources/out/pngs2/"

COMPRESSION_LEVEL = 49
BIT_RATE = 25
```

2.0.2 run

[3]:

```
! ffmpeg -i $VIDEO_PATH -vcodec libx264 -crf $COMPRESSION_LEVEL $VIDEO_OUTPUT
```

```
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '/home/jupyter/resources/video.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: isommp42
    creation_time   : 2019-12-23T23:25:30.000000Z
  Duration: 00:06:46.53, start: 0.000000, bitrate: 2050 kb/s
    Stream #0:0(und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv,
bt709), 1280x720 [SAR 1:1 DAR 16:9], 1919 kb/s, 25 fps, 25 tbr, 12800 tbn, 50
tbc (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> h264 (libx264))
  Stream #0:1 -> #0:1 (aac (native) -> aac (native))
Press [q] to stop, [?] for help
[libx264 @ 0x55a0d838dcc0] using SAR=1/1
[libx264 @ 0x55a0d838dcc0] using cpu capabilities: MMX2 SSE2Fast
SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[libx264 @ 0x55a0d838dcc0] profile High, level 3.1, 4:2:0, 8-bit
[libx264 @ 0x55a0d838dcc0] 264 - core 161 r3030M 8bd6d28 -
H.264/MPEG-4 AVC codec - Copyleft 2003-2020 - http://www.videolan.org/x264.html
```

```
- options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1
psy_rd=1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0
deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=12 lookahead_threads=2
sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0
constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1
open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0
rc_lookahead=40 rc=crf mbtree=1 crf=49.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4
ip_ratio=1.40 aq=1:1.00
Output #0, mp4, to '/home/jupyter/resources/out/vid2.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: isommp42
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: h264 (libx264) (avc1 / 0x31637661),
yuv420p(progressive), 1280x720 [SAR 1:1 DAR 16:9], q=-1--1, 25 fps, 12800 tbn,
25 tbc (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 aac
frame=10163 fps=151 q=-1.0 Lsize=   10816kB time=00:06:46.53 bitrate=
217.9kbits/s speed=6.03x
video:4119kB audio:6363kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: 3.181295%
[libx264 @ 0x55a0d838dcc0] frame I:43    Avg QP:44.59  size:  5549
[libx264 @ 0x55a0d838dcc0] frame P:2681  Avg QP:49.34  size:   927
[libx264 @ 0x55a0d838dcc0] frame B:7439  Avg QP:50.23  size:   201
[libx264 @ 0x55a0d838dcc0] consecutive B-frames:  0.8%  3.7%  3.5%
92.0%
[libx264 @ 0x55a0d838dcc0] mb I  I16..4: 25.2% 74.1%  0.7%
[libx264 @ 0x55a0d838dcc0] mb P  I16..4:  2.0%  2.3%  0.0%  P16..4:
11.6%  0.2%  0.3%  0.0%  0.0%    skip:83.6%
[libx264 @ 0x55a0d838dcc0] mb B  I16..4:  0.0%  0.0%  0.0%  B16..8:
4.9%  0.0%  0.0%  direct: 0.0%  skip:95.1%  L0:34.9% L1:65.0% BI: 0.0%
[libx264 @ 0x55a0d838dcc0] 8x8 transform intra:59.2% inter:98.8%
[libx264 @ 0x55a0d838dcc0] coded y,uvDC,uvAC intra: 4.9% 23.4% 0.2%
inter: 0.1% 0.4% 0.0%
```

```
[libx264 @ 0x55a0d838dcc0] i16 v,h,dc,p: 46% 37%  5% 11%
[libx264 @ 0x55a0d838dcc0] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 16% 10% 68%
1%  1%  1%  1%  1%  1%
[libx264 @ 0x55a0d838dcc0] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 25% 27% 19%
7%  5%  4%  7%  4%  3%
[libx264 @ 0x55a0d838dcc0] i8c dc,h,v,p: 96%  3%  2%  0%
[libx264 @ 0x55a0d838dcc0] Weighted P-Frames: Y:1.9% UV:0.9%
[libx264 @ 0x55a0d838dcc0] ref P L0: 71.7%  1.6% 17.6%  8.8%  0.2%
[libx264 @ 0x55a0d838dcc0] ref B L0: 90.6%  7.0%  2.4%
[libx264 @ 0x55a0d838dcc0] ref B L1: 96.5%  3.5%
[libx264 @ 0x55a0d838dcc0] kb/s:82.99
[aac @ 0x55a0d838b880] Qavg: 375.009
```

[33]:

```
! ffmpeg -i  $VIDEO_PATH -r $BIT_RATE -f image2 $IMAGE_OUTPUT
```

```
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '/home/jupyter/resources/video.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: isommp42
    creation_time   : 2019-12-23T23:25:30.000000Z
  Duration: 00:06:46.53, start: 0.000000, bitrate: 2050 kb/s
    Stream #0:0(und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv,
bt709), 1280x720 [SAR 1:1 DAR 16:9], 1919 kb/s, 25 fps, 25 tbr, 12800 tbn, 50
tbc (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to '/home/jupyter/resources/out/pngs/image-%07d.png':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: isommp42
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 60 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed= 2.4x
video:12561143kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
```

[ ]:

```
ffmpeg psnr ssim
```

[36]:

```
VIDEO_NAME = "video.mp4"
VIDEO_PATH = f"/home/jupyter/resources/{VIDEO_NAME}"
PATH = "/home/jupyter/resources/out/"
VIDEO_FOLDER = f"{PATH}video1"
BIT_RATE = 25
```

[31]:

```
VIDEO_NAME = "video.mp4"
VIDEO_PATH = f"/home/jupyter/resources/{VIDEO_NAME}"
PATH = "/home/jupyter/resources/out/"
VIDEO_FOLDER = f"{PATH}video1"

! mkdir -p $VIDEO_FOLDER

COMPRESSION_LEVEL = 49
for i in range(60, 70, 10):
    VIDEO_OUTPUT = f"{VIDEO_FOLDER}/vd_compresion_level{i}.mp4"
    COMPRESSION_LEVEL = i
    # print(VIDEO_OUTPUT)
    ! ffmpeg -i $VIDEO_PATH -vcodec libx264 -crf $COMPRESSION_LEVEL␣
 ↪$VIDEO_OUTPUT
```

```
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '/home/jupyter/resources/video.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: isommp42
    creation_time   : 2019-12-23T23:25:30.000000Z
  Duration: 00:06:46.53, start: 0.000000, bitrate: 2050 kb/s
    Stream #0:0(und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv,
bt709), 1280x720 [SAR 1:1 DAR 16:9], 1919 kb/s, 25 fps, 25 tbr, 12800 tbn, 50
tbc (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      creation_time   : 2019-12-23T23:25:30.000000Z
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
File '/home/jupyter/resources/out/video1/vd_compresion_level60.mp4' already
exists. Overwrite? [y/N] ^C
```

[ ]:

```
for i in range(60, 70, 10):
    COMPRESSION_LEVEL = i
```

```
! ffmpeg -i ./out/video1/vd_compresion_level10.mp4 -i video.mp4 -filter_complex␣
↪"psnr" -f null /dev/null


for i in range(60, 70, 10):
    COMPRESSION_LEVEL = i

! ffmpeg -i ./out/video1/vd_compresion_level10.mp4 -i video.mp4 -filter_complex␣
↪"SSIM" -f null /dev/null
```

[ ]:

```
PSNR y:50.766837 u:54.484554 v:53.982772 average:51.644865 min:48.443387 max:inf
```

[ ]:

```
PSNR y:45.242939 u:51.200485 v:50.461904 average:46.440201 min:42.440845 max:inf
```

[ ]:

```
PSNR y:39.074676 u:47.472061 v:46.372864 average:40.490389 min:37.249966 max:inf
```

[ ]:

```
PSNR y:33.037487 u:43.827164 v:42.499419 average:34.590061 min:31.660728 max:inf
```

[ ]:

```
PSNR y:27.019074 u:40.451990 v:38.771928 average:28.659893 min:25.099286 max:inf
```

[ ]:

```
PSNR y:26.543305 u:40.037757 v:38.245385 average:28.183968 min:24.637782 max:61.
↪100454
```

[29]:

```
psnr_average = [51.644865, 46.440201, 40.490389, 34.590061, 28.659893, 28.
↪183968]
```

[ ]:

```

```

[ ]:

```
SSIM Y:0.996373 (24.404208) U:0.996611 (24.699032) V:0.996550 (24.621753) All:0.
↪996442 (24.487880)
```

[ ]:

```
SSIM Y:0.990121 (20.052841) U:0.993226 (21.691724) V:0.992785 (21.417548) All:0.
↪991082 (20.497558)
```

[ ]:

```
SSIM Y:0.970103 (15.243780) U:0.986569 (18.719019) V:0.984363 (18.058591) All:0.
↪975224 (16.059755)
```

[ ]:

```
SSIM Y:0.903795 (10.168004) U:0.977690 (16.514917) V:0.970261 (15.266734) All:0.
↪927188 (11.377978)
```

[ ]:

```
SSIM Y:0.757815 (6.158522) U:0.965733 (14.651223) V:0.946255 (12.696606) All:0.
↪823874 (7.541776)
```

[ ]:

```
SSIM Y:0.743938 (5.916553) U:0.963808 (14.413909) V:0.942110 (12.373999) All:0.
↪813612 (7.295819)
```

[ ]:

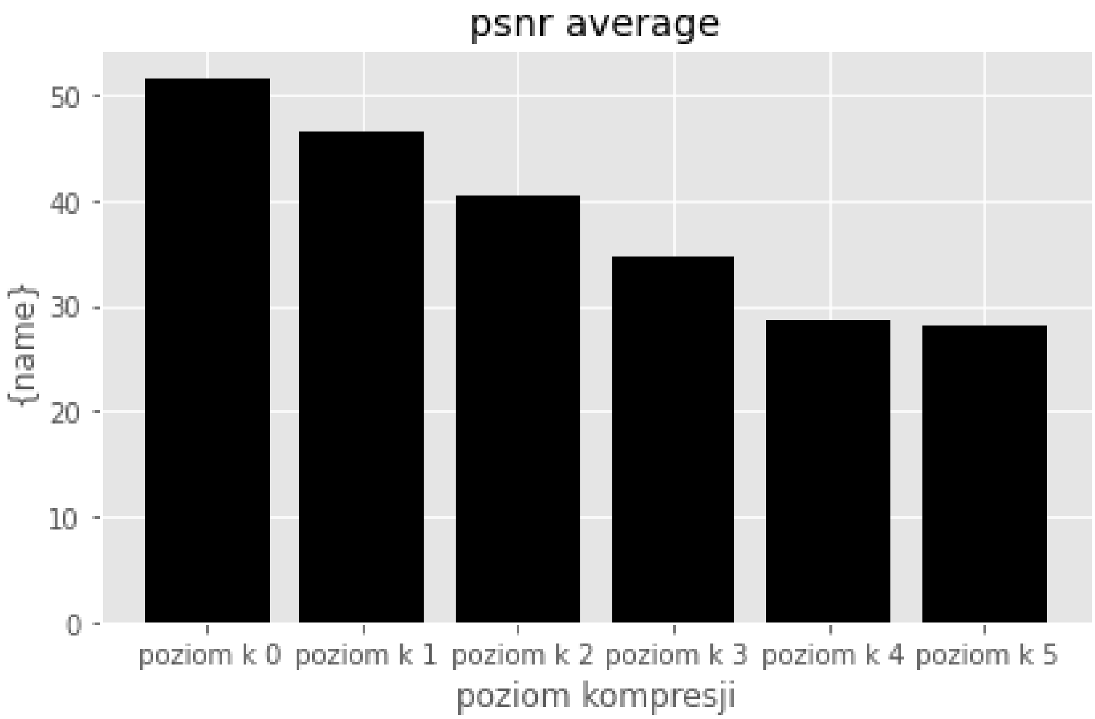
```

```

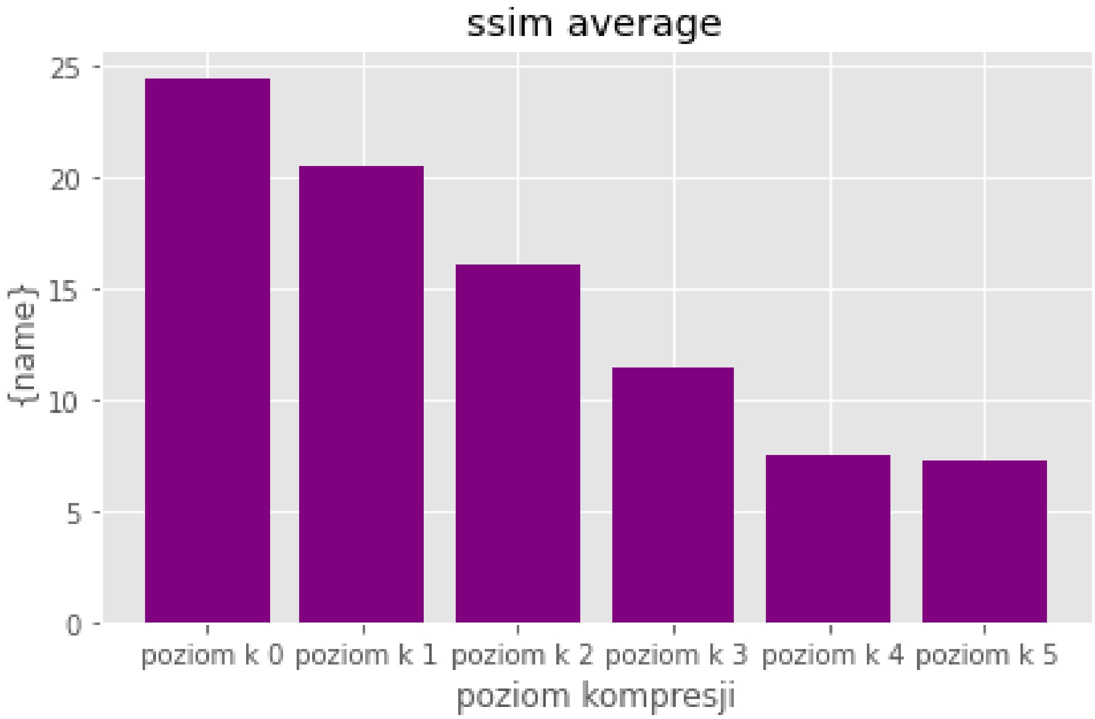
[32]:

```
ssim_average = [24.487880, 20.497558, 16.059755, 11.377978, 7.541776, 7.295819]
```

[34]:

```
import matplotlib.pyplot as plt
import random
%matplotlib inline
plt.style.use('ggplot')

NUMBER=6

x = [f"  poziom k {i} " for i in range(NUMBER)]


def show_chart( lables, values, name, col="green"  ):
    plt.bar(x_pos, values, color=f"{col}")
    plt.xlabel("poziom kompresji")
    plt.ylabel("{name}")
    plt.title(f"{name} average")
    plt.xticks(x_pos, x)
    plt.tight_layout()
    plt.show()

x_pos = [i for i in range(NUMBER)]

color = "blue"
colour = [ "red", "blue", "green", "yellow", "purple", "orange", "white",␣
 ↪"black" ]

show_chart(x_pos, psnr_average, "psnr", random.choice(colour))
show_chart(x_pos, ssim_average, "ssim", random.choice(colour))
```


psnr average
{name}
0
10
20
30
40
50
poziom k 0
poziom k 1
poziom k 2
poziom k 3
poziom k 4
poziom k 5
poziom kompresji


ssim average
{name}
0
5
10
15
20
25
poziom k 0
poziom k 1
poziom k 2
poziom k 3
poziom k 4
poziom k 5
poziom kompresji

[29]:

```python
import glob
import os.path

files = glob.glob(os.path.join(VIDEO_FOLDER, '*.mp4'))

BIT_RATE = 25


for i, file in enumerate(files):
    VIDEO_PATH = str(file)
    INDEX = i*10
    IMAGE_OUTPUT = f"{VIDEO_FOLDER}/pngs_compresion_level{INDEX}/image-%07d.png"
    # print(f"ffmpeg -i  {VIDEO_PATH} -r {BIT_RATE} -f image2 {IMAGE_OUTPUT}")
    print(f"mkdir -p {VIDEO_FOLDER}/pngs_compresion_level{i*10}")
    ! mkdir -p $VIDEO_FOLDER/pngs_compresion_level$INDEX
    ! ffmpeg -i  $VIDEO_PATH -r $BIT_RATE -f image2 $IMAGE_OUTPUT
```

```
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level0
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level60.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 200 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 65 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level0/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 66 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=2.65x
video:9643557kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level10
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level20.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 2451 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 2316 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level10/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 52 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=2.07x
video:12841216kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level20
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level50.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 208 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 73 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level20/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 64 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=2.57x
video:9848475kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level30
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
```

```
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level40.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 359 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 224 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level30/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 57 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=2.27x
```

```
video:11265374kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level40
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level30.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 840 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 705 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level40/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 52 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed= 2.1x
video:12197487kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
mkdir -p /home/jupyter/resources/out/video1/pngs_compresion_level50
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level10.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 8162 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 8028 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level50/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 18 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=0.729x
video:13164859kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
```

[37]:

```
VIDEO_PATH = f"{VIDEO_FOLDER}/vd_compresion_level60.mp4"
IMAGE_OUTPUT = f"{VIDEO_FOLDER}/pngs_compresion_level60/image-%07d.png"
! ffmpeg -i  $VIDEO_PATH -r $BIT_RATE -f image2 $IMAGE_OUTPUT
```

```
ffmpeg version 9c33b2f Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 9.3.0 (crosstool-NG 1.24.0.133_b0863d8_dirty)
  configuration: --prefix=/opt/conda --cc=/home/conda/feedstock_root/build_artif
acts/ffmpeg_1627813612080/_build_env/bin/x86_64-conda-linux-gnu-cc --disable-doc
--disable-openssl --enable-avresample --enable-gnutls --enable-gpl --enable-
hardcoded-tables --enable-libfreetype --enable-libopenh264 --enable-libx264
--enable-pic --enable-pthreads --enable-shared --enable-static --enable-version3
--enable-zlib --enable-libmp3lame --pkg-config=/home/conda/feedstock_root/build_
artifacts/ffmpeg_1627813612080/_build_env/bin/pkg-config
  libavutil      56. 51.100 / 56. 51.100
  libavcodec     58. 91.100 / 58. 91.100
  libavformat    58. 45.100 / 58. 45.100
  libavdevice    58. 10.100 / 58. 10.100
  libavfilter     7. 85.100 /  7. 85.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  7.100 /  5.  7.100
  libswresample   3.  7.100 /  3.  7.100
  libpostproc    55.  7.100 / 55.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'/home/jupyter/resources/out/video1/vd_compresion_level60.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
  Duration: 00:06:46.56, start: 0.000000, bitrate: 200 kb/s
    Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720
[SAR 1:1 DAR 16:9], 65 kb/s, 25 fps, 25 tbr, 12800 tbn, 50 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo,
fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to
'/home/jupyter/resources/out/video1/pngs_compresion_level60/image-%07d.png':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.45.100
    Stream #0:0(und): Video: png, rgb24, 1280x720 [SAR 1:1 DAR 16:9], q=2-31,
200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc. Created on:
12/23/2019.
      encoder         : Lavc58.91.100 png
frame=10163 fps= 65 q=-0.0 Lsize=N/A time=00:06:46.52 bitrate=N/A speed=2.61x
video:9643557kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: unknown
```

[43]:

```
PATH = "/home/jupyter/resources/out/video1"
rang = "image-0003300.png image-0003384.png"
IMG_NAME = "image-000"
EXT = ".png"


for i in range(10, 70, 10):
    COMPRESION_INDEX = i
    ! mkdir -p /home/jupyter/resources/out/video1/
 ↪samle_pngs_compresion_level$COMPRESION_INDEX
    for j in range(3300, 3384):
        IMG_INDEX = j
        ! cp /home/jupyter/resources/out/video1/
 ↪pngs_compresion_level$COMPRESION_INDEX/$IMG_NAME$IMG_INDEX$EXT /home/jupyter/
 ↪resources/out/video1/samle_pngs_compresion_level$COMPRESION_INDEX/
```

```
# ! cd $path
# ! for i in {1..6}; mkdir -p $PATH$same_pngs_compresion_level$()
# ! for (( k = 10; k < 70; k+=10 )); do cp /home/jupyter/resources/out/video1/
↪image-0003300.png; echo "$a"; done
```

2.0.3 setup code

[9]:

```
! pip install imutils

import cv2
import imutils
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import ImageGrid
import typing
from typing import Tuple
from dataclasses import dataclass
import dlib
from dlib import rectangle

import itertools


@dataclass
class Face_IMG:
    label: str
    left: typing.Tuple[int, int]
    top: typing.Tuple[int, int]
    right: typing.Tuple[int, int]
    bottom: Tuple[int, int]

def custom_imread(my_path, img_width=None, img_grayscale=False):

  image_out = cv2.imread(my_path, 0 if img_grayscale else 1)
  if img_width:
    image_out = imutils.resize(image_out, width=img_width, inter=cv2.
↪INTER_CUBIC)

  image_out = cv2.cvtColor(image_out, cv2.COLOR_BGR2RGB)

  return image_out

def custom_imshow(*args, figsize_img=(12,8)):

  figure = plt.figure(figsize=figsize_img)
  cols_img = 2 if len(args) > 1 else 1
  rows_nr = max(round(len(args)/2), 1)
  grid_mesh = ImageGrid(figure, 111, nrows_ncols=(rows_nr, cols_img),␣
↪axes_pad=0.1)

  for i, j in zip(grid_mesh, args):
    i.axis('off')
    i.imshow(j)

  plt.show()

def draw_board_detected(img, img_faces: typing.List[Face_IMG]):
  label_colors = {}
  colors = itertools.cycle([(255, 255, 255), [0,255,255], [255,0,255],␣
↪[255,0,255]])
  img_draw = img.copy()
  for i, face in enumerate(img_faces):
    if not face.label in label_colors.keys():
      label_colors[face.label] = next(colors)
    cv2.rectangle(img_draw, (face.left, face.top), (face.right, face.bottom),␣
↪label_colors[face.label], 6)

  height, _, _ = img_draw.shape
  text_margin_left = 20
  for (label, color) in label_colors.items():
    (text_w, _) = draw_text(img_draw, label, pos=(text_margin_left, height-40),␣
↪text_color=color)
    text_margin_left += 20 + text_w

  return img_draw

def draw_text(img, text,
          font=cv2.FONT_HERSHEY_SIMPLEX,
          pos=(0, 0),
          font_scale=1,
          font_thickness=2,
          text_color=(0, 255, 0),
          text_color_bg=(0, 0, 0),
          padding=5
          ):

    x, y = pos
    text_size, _ = cv2.getTextSize(text, font, font_scale, font_thickness)
    text_w, text_h = text_size
    cv2.rectangle(img, (x - padding, y - padding), (x + text_w + padding, y +␣
↪text_h + padding), text_color_bg, -1)
    cv2.putText(img, text, (x, y + text_h + font_scale - 1), font, font_scale,␣
↪text_color, font_thickness)
```

```
    return (text_w + padding, text_h + padding)
```

```
Requirement already satisfied: imutils in /opt/conda/lib/python3.7/site-packages
(0.5.4)
```

2.0.4 setup vars

[128]:
```
img_base = custom_imread("/home/jupyter/resources/out/pngs/image-0002199.png",␣
 ↪img_width=800)
custom_imshow(img)
```

2.0.5 declarations

[3]:
```
face_cascade = cv2.CascadeClassifier('/home/jupyter/src/
 ↪haarcascade_frontalface_default.xml')


def face_detect_haar(img, scaleFactor=10, minNeighbors=3, minSize=(10, 10)):
  faces = face_cascade.detectMultiScale(
    img,
    scaleFactor=scaleFactor,    # specifying how much the image size is reduced␣
 ↪at each image scale
    minNeighbors=minNeighbors, # specifying how many neighbors each candidate␣
 ↪rectangle should have to retain it
```

```
    minSize=minSize,             # minimum possible object size. Objects smaller␣
 ↪than that are ignored.
  )
  return [Face_IMG("HAAR", x, y, x+w, y+h) for (x,y,w,h) in faces]
```

[4]:

```
scaleFactor = 2

# docs: http://dlib.net/python/index.html#dlib.get_frontal_face_detector
face_detect = dlib.get_frontal_face_detector()

# def face_detect_hog(img, return_list_hog):
def face_detect_hog(img):
  dets = face_detect(img, scaleFactor)
  return [Face_IMG('HOG', d.left(), d.top(), d.right(), d.bottom()) for d in␣
 ↪dets]
```

[7]:

```
# !wget -O mmod_human_face_detector.dat.bz2 http://dlib.net/files/
 ↪mmod_human_face_detector.dat.bz2
# !bzip2 -d mmod_human_face_detector.dat.bz2

scaleFactor = 3

# docs: http://dlib.net/python/index.html#dlib.cnn_face_detection_model_v1
cnn_face_detect = dlib.cnn_face_detection_model_v1('mmod_human_face_detector.
 ↪dat')

def face_detect_dnn(img):
  dets = cnn_face_detect(img, scaleFactor)
  return [Face_IMG('DNN', d.rect.left(), d.rect.top(), d.rect.right(), d.rect.
 ↪bottom()) for d in dets]

img = custom_imread("/home/jupyter/resources/out/video1/
 ↪samle_pngs_compresion_level10/image-0003344.png", img_width=800)
custom_imshow(draw_board_detected(img, face_detect_dnn(img) ))
```

```
[174]: face_cascade = cv2.CascadeClassifier('/home/jupyter/src/
        ↪haarcascade_frontalface_default.xml')


       def face_detect_haar(img, scaleFactor=10, minNeighbors=2, minSize=(1, 1)):
         faces = face_cascade.detectMultiScale(
           img,
           scaleFactor=scaleFactor,   # specifying how much the image size is reduced␣
        ↪at each image scale
           minNeighbors=minNeighbors, # specifying how many neighbors each candidate␣
        ↪rectangle should have to retain it
           minSize=minSize,            # minimum possible object size. Objects smaller␣
        ↪than that are ignored.
         )
         return [Face_IMG("HAAR", x, y, x+w, y+h) for (x,y,w,h) in faces]

       img = custom_imread("/home/jupyter/resources/out/video1/
        ↪samle_pngs_compresion_level10/image-0003344.png", img_width=800)
       custom_imshow(draw_board_detected(img, face_detect_haar(img) ))
       # print(len(face_detect_dnn(img_base)))
```

[21]:

```
import multiprocessing as mp
import time
import os
from os import listdir
from os.path import isfile, join

# scaleFactor = 3
# minNeighbors = 3
# minSize = 10

mypath="/home/jupyter/resources/out/pngs2/"
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]


result_list = []
def log_result(result):
    result_list.append(result)

max_pool=7
```

[ ]:

```
pool = mp.Pool(max_pool)

rs = []
for i, img in enumerate(onlyfiles):
    img = custom_imread(os.path.join(mypath, img), img_width=800)
    r = pool.apply_async(face_detect_hog, args = (img, ), callback = log_result)
    rs.append(r)

for r in rs:
    r.wait()
pool.close()
pool.join()

hog_results = result_list
result_list = []
```

[12]:

```
pool = mp.Pool(max_pool)

rs = []
for i, img in enumerate(onlyfiles):
    img = custom_imread(os.path.join(mypath, img), img_width=800)
    r = pool.apply_async(face_detect_haar, args = (img, ), callback =␣
 ↪log_result)
    rs.append(r)

for r in rs:
    r.wait()
pool.close()
pool.join()

haar_results = result_list
result_list = []
```

[22]:

```
pool = mp.Pool(max_pool)

rs = []
for i, img in enumerate(onlyfiles):
    img = custom_imread(os.path.join(mypath, img), img_width=800)
    r = pool.apply_async(face_detect_dnn, args = (img, ), callback = log_result)
    rs.append(r)

for r in rs:
    r.wait()
pool.close()
pool.join()

dnn_results = result_list
result_list = []
```

[1]:

```
import os
from os import listdir
from os.path import isfile, join
import pickle
import glob
import os.path

mypath="/home/jupyter/resources/out/pngs2/"
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]

PATH = "/home/jupyter/resources/out/video1"
rang = "image-0003300.png image-0003384.png"
IMG_NAME = "image-000"
EXT = ".png"

! mkdir -p /home/jupyter/pickles/


for i in range(10, 70, 10):
    COMPRESION_INDEX = i
    PNGS = f"/home/jupyter/resources/out/video1/
 ↪samle_pngs_compresion_level{COMPRESION_INDEX}"
    onlyfiles = [f for f in listdir(PNGS) if isfile(join(PNGS, f))]

    try:
        results_hog = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_hog{COMPRESION_INDEX}", "rb"))

    except (OSError, IOError) as e:
        if "results_hog" in str(e):
            results_hog = []
            for i, img in enumerate(onlyfiles):
                # print("hog", i)
                imm = img
                img = custom_imread(os.path.join(PNGS, img), img_width=800)
                results_hog.append( face_detect_hog(img) )
                pickle.dump(results_hog, open(f"/home/jupyter/pickles/
 ↪results_hog{COMPRESION_INDEX}", "wb"))
        else:
            print("error: ", e)

    try:
        results_haar = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_haar{COMPRESION_INDEX}", "rb"))

    except (OSError, IOError) as e:
        if "results_haar" in str(e):
            results_haar = []
            for i, img in enumerate(onlyfiles):
                # print("harr", i)
                imm = img
                img = custom_imread(os.path.join(PNGS, img), img_width=800)
                results_haar.append( face_detect_haar(img) )
                pickle.dump(results_haar, open(f"/home/jupyter/pickles/
↪results_haar{COMPRESION_INDEX}", "wb"))
        else:
            print("error: ", e)

    try:
        results_dnn = pickle.load(open(f"/home/jupyter/pickles/
↪results_dnn{COMPRESION_INDEX}", "rb"))
    except (OSError, IOError) as e:
        if "results_dnn" in str(e):
            results_dnn = []
            for i, img in enumerate(onlyfiles):
                # print("dnn", i)
                imm = img
                img = custom_imread(os.path.join(PNGS, img), img_width=800)
                results_dnn.append( face_detect_dnn(img) )
                pickle.dump(results_dnn, open(f"/home/jupyter/pickles/
↪results_dnn{COMPRESION_INDEX}", "wb"))
        else:
            print("error: ", e)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipykernel_3151/892810402.py in <module>
     23 
     24     try:
---> 25         results_hog = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_hog{COMPRESION_INDEX}", "rb"))
     26 
     27     except (OSError, IOError) as e:

AttributeError: Can't get attribute 'Face_IMG' on <module '__main__'>
```

[10]:

```
import pickle

levels_hog = []
levels_haar = []
levels_dnn = []

recognition_results = {
    "results_hog": levels_hog,
    "results_haar": levels_haar,
    "results_dnn": levels_dnn,
}
```

```python

def results_cleanup(results):
    return [result for result in results if len(result) is not 0]

for i in range(10, 70, 10):
    COMPRESION_INDEX = i
    VIDEO_FOLDER = f"/home/jupyter/resources/out/video1/
 ↪samle_pngs_compresion_level{COMPRESION_INDEX}"
    results_hog = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_hog{COMPRESION_INDEX}", "rb"))
    results_haar = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_haar{COMPRESION_INDEX}", "rb"))
    results_dnn = pickle.load(open(f"/home/jupyter/pickles/
 ↪results_dnn{COMPRESION_INDEX}", "rb"))

    results_hog = results_cleanup(results_hog)
    levels_hog.append(results_hog)

    results_haar = results_cleanup(results_haar)
    levels_haar.append(results_haar)

    results_dnn = results_cleanup(results_dnn)
    levels_dnn.append(results_dnn)

# print(recognition_results.get("results_haar"))
```

[16]:

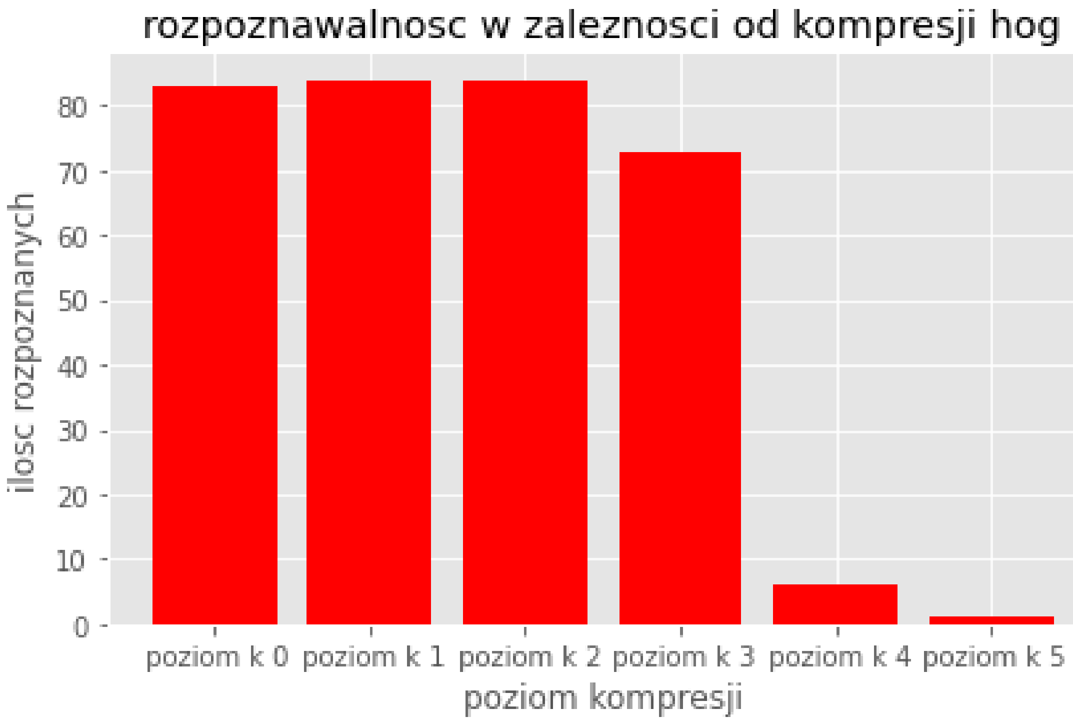
```python
print("hog")
results_len_hog = [len(i) for i in recognition_results.get("results_hog")]
print(results_len_hog)

print("haar")
results_len_haar = [len(i) for i in recognition_results.get("results_haar")]

print("dnn")
results_len_dnn = [len(i) for i in recognition_results.get("results_dnn")]
print(results_len_dnn)
```

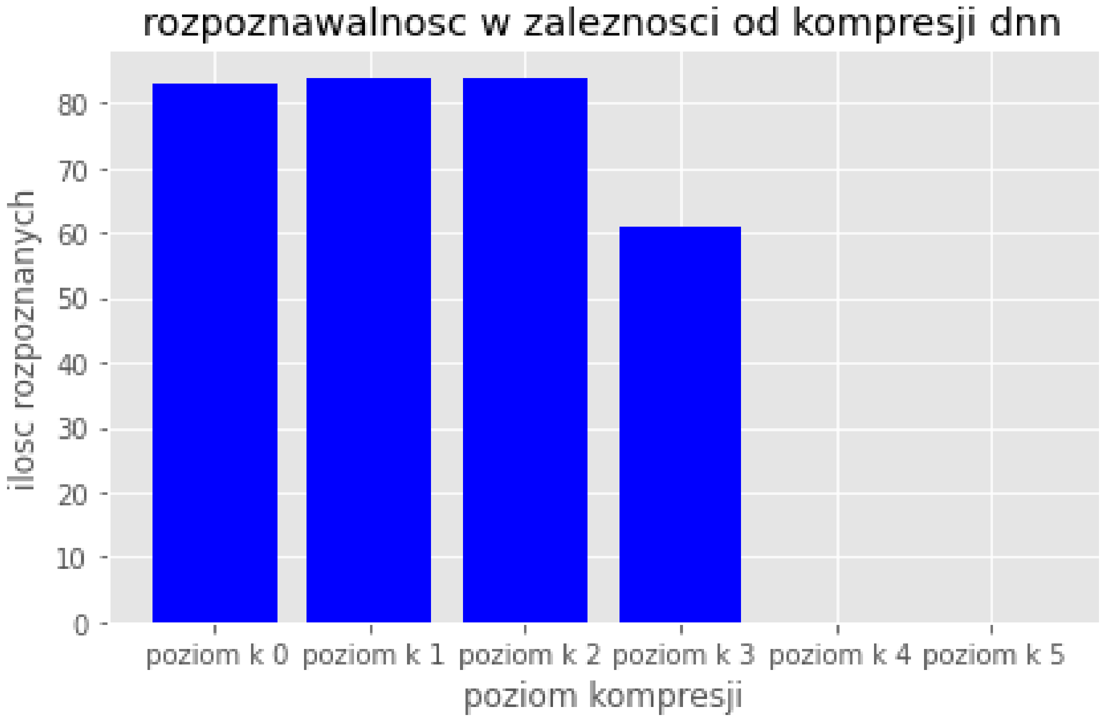
```
hog
[83, 84, 84, 73, 6, 1]
haar
dnn
[83, 84, 84, 61, 0, 0]
```

[27]:

```python
import matplotlib.pyplot as plt
import random
%matplotlib inline
plt.style.use('ggplot')
```

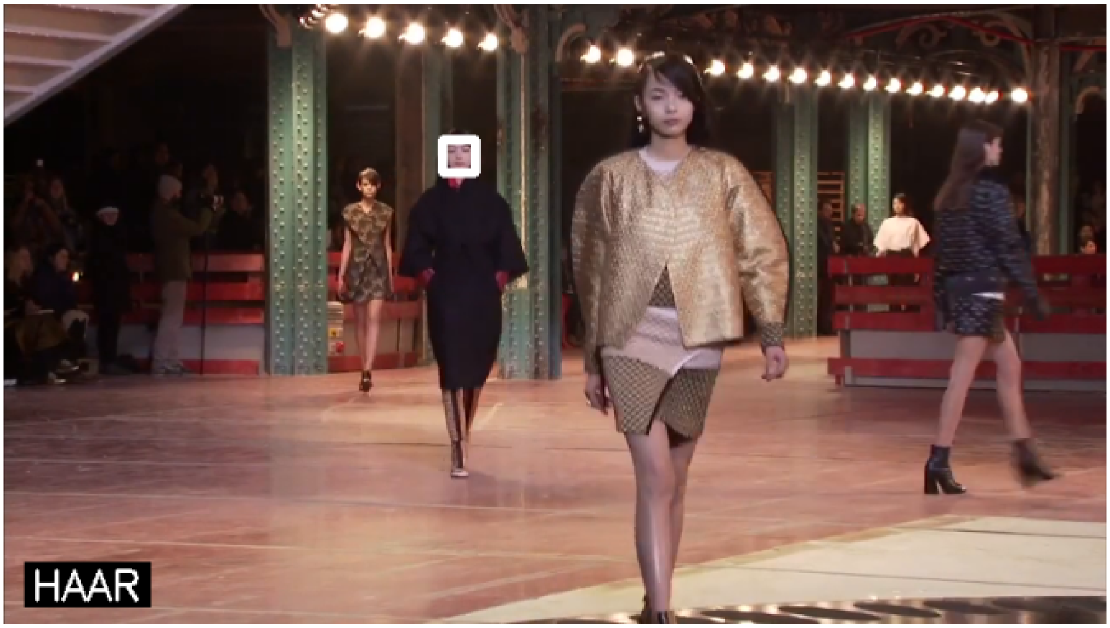
```
NUMBER=6

x = [f"  poziom k {i} " for i in range(NUMBER)]


def show_chart( lables, values, name, col="green"  ):
    plt.bar(x_pos, values, color=f"{col}")
    plt.xlabel("poziom kompresji")
    plt.ylabel("ilosc rozpoznanych")
    plt.title(f"rozpoznawalnosc w zaleznosci od kompresji {name}")
    plt.xticks(x_pos, x)
    plt.tight_layout()
    plt.show()

x_pos = [i for i in range(NUMBER)]

color = "blue"
colour = [ "red", "blue", "green", "yellow", "purple", "orange", "white",␣
 ↪"black" ]

show_chart(x_pos, results_len_hog, "hog", random.choice(colour))
# show_chart(x_pos, results_len_hog, "haar", random.choice(colour))
show_chart(x_pos, results_len_dnn, "dnn", random.choice(colour))
```

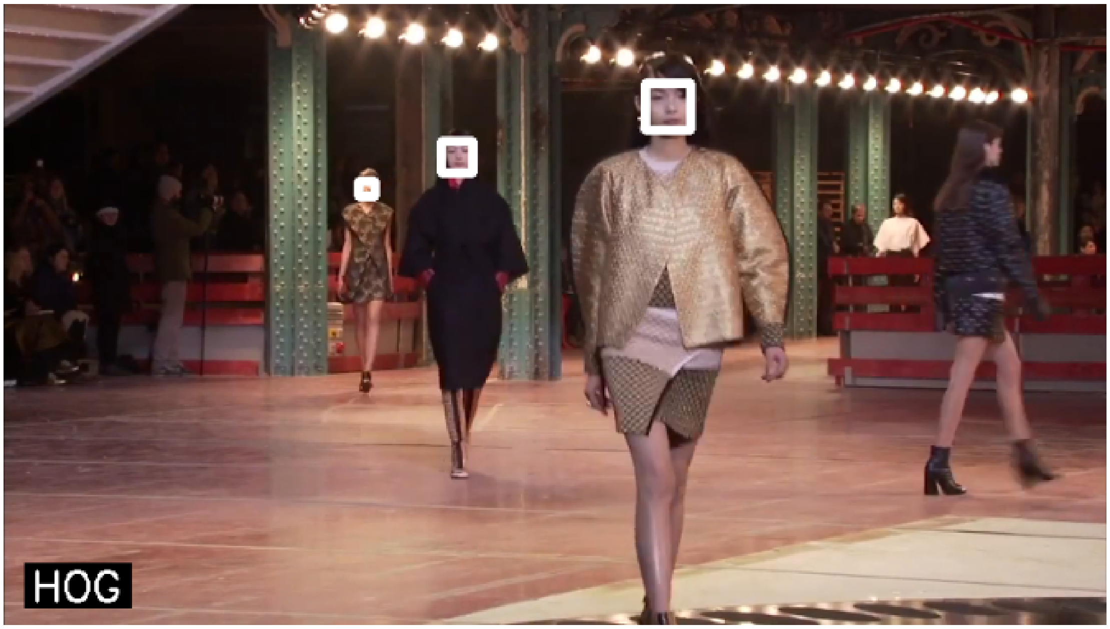


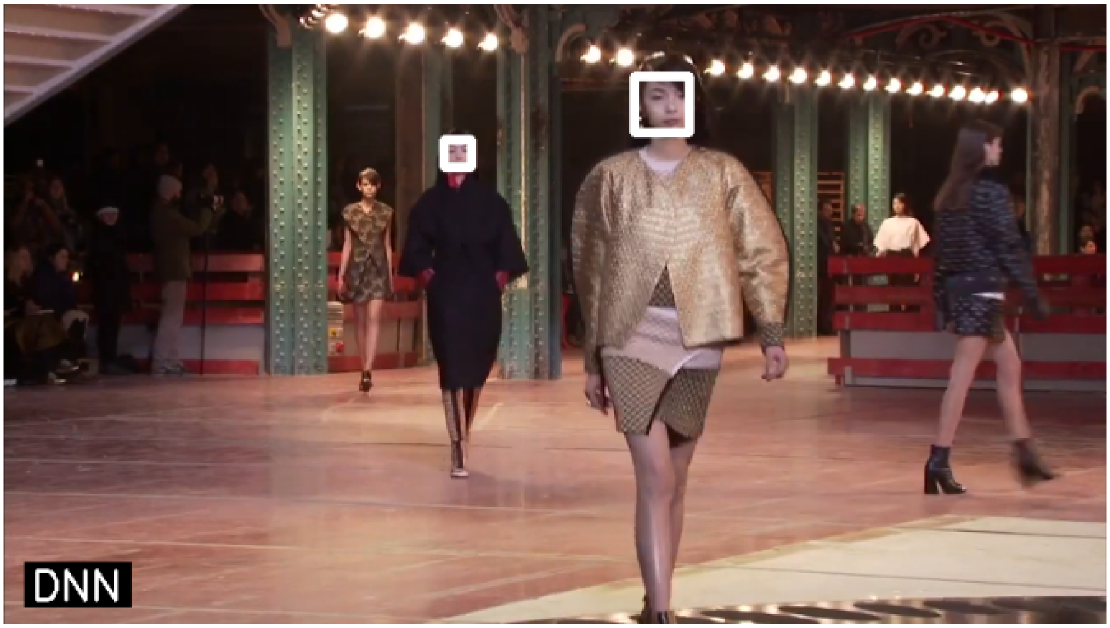


2.0.6 img_show

[52]:
```
custom_imshow(draw_board_detected(img, faces_haar))
custom_imshow(draw_board_detected(img, faces_hog))
custom_imshow(draw_board_detected(img, faces_dnn))
```


HAAR


HOG



2.0.7 quality detection

declarations

[25]:
```
from collections import defaultdict
from scipy.stats import itemfreq
from scipy import ndimage as ndi
import matplotlib.pyplot as plt
from skimage import feature
from PIL import Image as IMG
import numpy as np
import pandas as pd
import operator
import cv2
import os

from IPython.core.display import HTML
from IPython.display import Image
import glob
import os.path




VIDEO_NAME = "video.mp4"
VIDEO_PATH = f"/home/jupyter/resources/{VIDEO_NAME}"
PATH = "/home/jupyter/resources/out/"
```

```
VIDEO_FOLDER = f"{PATH}video1"

features = []
for i, j in zip( range(10, 70, 10), range(0, 6) ):
    COMPRESSION_LEVEL = i
    features.append( pd.DataFrame() )
    imgs = f"{VIDEO_FOLDER}/samle_pngs_compresion_level{COMPRESSION_LEVEL}"
    files = glob.glob(os.path.join(f"{VIDEO_FOLDER}/
 ↪samle_pngs_compresion_level{COMPRESSION_LEVEL}", '*.png'))
    # print(files)
    features[j]['image'] = files

# features = pd.DataFrame()
# features['image'] = imgs
```

[36]:

```
def color_analysis(img):
    # obtain the color palatte of the image
    palatte = defaultdict(int)
    for pixel in img.getdata():
        palatte[pixel] += 1

    # sort the colors present in the image
    sorted_x = sorted(palatte.items(), key=operator.itemgetter(1), reverse =␣
 ↪True)
    light_shade, dark_shade, shade_count, pixel_limit = 0, 0, 0, 25
    for i, x in enumerate(sorted_x[:pixel_limit]):
        if all(xx <= 20 for xx in x[0][:3]): ## dull : too much darkness
            dark_shade += x[1]
        if all(xx >= 240 for xx in x[0][:3]): ## bright : too much whiteness
            light_shade += x[1]
        shade_count += x[1]

    light_percent = round((float(light_shade)/shade_count)*100, 2)
    dark_percent = round((float(dark_shade)/shade_count)*100, 2)
    return light_percent, dark_percent
```

[35]:

```
def perform_color_analysis(img, flag):
    path =  img
    im = IMG.open(path) #.convert("RGB")

    # cut the images into two halves as complete average may give bias results
    size = im.size
    halves = (size[0]/2, size[1]/2)
    im1 = im.crop((0, 0, size[0], halves[1]))
    im2 = im.crop((0, halves[1], size[0], size[1]))

    try:
        light_percent1, dark_percent1 = color_analysis(im1)
        light_percent2, dark_percent2 = color_analysis(im2)
    except Exception as e:
        print(e)
        return None

    light_percent = (light_percent1 + light_percent2)/2
    dark_percent = (dark_percent1 + dark_percent2)/2

    if flag == 'black':
        return dark_percent
    elif flag == 'white':
        return light_percent
    else:
        return None
```

[34]:

```
def average_pixel_width(img):
    path =  img
    im = IMG.open(path)
    im_array = np.asarray(im.convert(mode='L'))
    edges_sigma1 = feature.canny(im_array, sigma=3)
    apw = (float(np.sum(edges_sigma1)) / (im.size[0]*im.size[1]))
    return apw*100
```

[33]:

```
def get_dominant_color(img):
    path =  img
    img = cv2.imread(path)
    arr = np.float32(img)
    pixels = arr.reshape((-1, 3))

    n_colors = 5
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 200, .1)
    flags = cv2.KMEANS_RANDOM_CENTERS
    _, labels, centroids = cv2.kmeans(pixels, n_colors, None, criteria, 10,␣
↪flags)

    palette = np.uint8(centroids)
    quantized = palette[labels.flatten()]
    quantized = quantized.reshape(img.shape)

    dominant_color = palette[np.argmax(itemfreq(labels)[:, -1])]
    return dominant_color
```

[32]:

```
def get_blurrness_score(image):
    path =  image
    image = cv2.imread(path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    fm = cv2.Laplacian(image, cv2.CV_64F).var()
    return fm
```

results

[ ]:

```
for i in range(0, 6):
    features[i]['dullness'] = features[i]['image'].apply(lambda x :␣
 ↪perform_color_analysis(x, 'black'))
    topdull = features[i].sort_values('dullness', ascending = False)
    features[i]['whiteness'] = features[i]['image'].apply(lambda x :␣
 ↪perform_color_analysis(x, 'white'))
    topdull = features[i].sort_values('whiteness', ascending = False)
    features[i]['average_pixel_width'] = features[i]['image'].
 ↪apply(average_pixel_width)
    tempdf = features[i].sort_values('average_pixel_width').head()
    features[i]['dominant_color'] = features[i]['image'].
 ↪apply(get_dominant_color)
    features[i]['blurrness'] = features[i]['image'].apply(get_blurrness_score)
    features[i][['image','blurrness']].head(5)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:16:
DeprecationWarning:     `itemfreq` is deprecated!
    `itemfreq` is deprecated and will be removed in a future version. Use
instead `np.unique(…, return_counts=True)`
  app.launch_new_instance()
```

[136]:

```
pickle.dump(features, open(f"/home/jupyter/pickles/results_features", "wb"))
```

[138]:

```
features = pickle.load(open(f"/home/jupyter/pickles/results_features", "rb"))
```

[ ]:

```
try:
    results_haar = pickle.load(open(f"/home/jupyter/pickles/results_features",␣
 ↪"rb"))

except (OSError, IOError) as e:

            pickle.dump(results_haar, open(f"/home/jupyter/pickles/
 ↪results_haar{COMPRESION_INDEX}", "wb"))
    else:
        print("error: ", e)
```

[8]:

```
features['dullness'] = features['image'].apply(lambda x :␣
 ↪perform_color_analysis(x, 'black'))
topdull = features.sort_values('dullness', ascending = False)
topdull.head(5)
```

```
---------------------------------------------------------------------------
```

```
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_3144/2835292562.py in <module>
----> 1 features['dullness'] = features['image'].apply(lambda x :␣
 ↪perform_color_analysis(x, 'black'))
      2 topdull = features.sort_values('dullness', ascending = False)
      3 topdull.head(5)

TypeError: list indices must be integers or slices, not str
```

```
[13]: features['whiteness'] = features['image'].apply(lambda x :␣
       ↪perform_color_analysis(x, 'white'))
      topdull = features.sort_values('whiteness', ascending = False)
      topdull.head(5)
```

```
[13]:                image  dullness  whiteness
      8  image-0010144.png    98.460      1.540
      2  image-0010145.png    98.510      1.490
      0  image-0010146.png    98.595      1.405
      9  image-0010147.png    98.685      1.315
      3  image-0010148.png    97.710      1.295
```

```
[14]: features['average_pixel_width'] = features['image'].apply(average_pixel_width)
      tempdf = features.sort_values('average_pixel_width').head()
      tempdf
```

```
[14]:                image  dullness  whiteness  average_pixel_width
      1  image-0010153.png    93.235      0.000             0.424913
      7  image-0010152.png    97.280      0.000             0.535590
      4  image-0010151.png    98.155      0.000             0.990668
      6  image-0010150.png    98.840      1.160             1.300130
      5  image-0010149.png    96.800      1.235             1.461480
```

```
[50]: features['dominant_red'] = features['dominant_color'].apply(lambda x: x[0]) /␣
       ↪255
      features['dominant_green'] = features['dominant_color'].apply(lambda x: x[1]) /␣
       ↪255
      features['dominant_blue'] = features['dominant_color'].apply(lambda x: x[2]) /␣
       ↪255
      features[['dominant_red', 'dominant_green', 'dominant_blue']].head(5)
```

```
[50]:    dominant_red  dominant_green  dominant_blue
      0      0.031373        0.019608       0.027451
      1      0.011765        0.007843       0.011765
      2      0.031373        0.019608       0.027451
      3      0.023529        0.015686       0.023529
      4      0.019608        0.011765       0.015686
```

[15]:
```
features['dominant_color'] = features['image'].apply(get_dominant_color)
features.head(10)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:16:
DeprecationWarning:     `itemfreq` is deprecated!
    `itemfreq` is deprecated and will be removed in a future version. Use
instead `np.unique(…, return_counts=True)`
  app.launch_new_instance()
```

[15]:
```
               image  dullness  whiteness  average_pixel_width dominant_color
0  image-0010146.png    98.595      1.405             1.986762      [8, 5, 7]
1  image-0010153.png    93.235      0.000             0.424913      [3, 2, 3]
2  image-0010145.png    98.510      1.490             2.048177      [8, 5, 7]
3  image-0010148.png    97.710      1.295             1.635959      [6, 4, 6]
4  image-0010151.png    98.155      0.000             0.990668      [5, 3, 4]
5  image-0010149.png    96.800      1.235             1.461480      [6, 4, 5]
6  image-0010150.png    98.840      1.160             1.300130      [5, 4, 5]
7  image-0010152.png    97.280      0.000             0.535590      [4, 2, 3]
8  image-0010144.png    98.460      1.540             2.223850      [8, 5, 8]
9  image-0010147.png    98.685      1.315             1.724175      [7, 5, 6]
```

[16]:
```
features['blurrness'] = features['image'].apply(get_blurrness_score)
features[['image','blurrness']].head(5)
```

[16]:
```
               image    blurrness
0  image-0010146.png  1058.767929
1  image-0010153.png   121.190068
2  image-0010145.png  1062.820085
3  image-0010148.png  1054.088450
4  image-0010151.png   153.176543
```

[139]:
```
features[0].head(10)
```

[139]:
```
                                               image dominant_color  devide  \
0  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
1  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
2  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
3  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
4  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
5  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
6  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
7  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 33]      10
8  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
9  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10

   blurrness2  blurrness2_devide  average_pixel_width_devide  dullness_devide  \
0    9.260924                 10                          10                2
```

```
1    9.509223                    10                           10                 2
2    8.217522                    10                           10                 2
3    9.176203                    10                           10                 2
4    8.920611                    10                           10                 2
5    8.949320                    10                           10                 2
6    8.074758                    10                           10                 2
7    9.925857                    10                           10                 2
8    9.740959                    10                           10                 2
9    9.381716                    10                           10                 2

   dullness2  average_pixel_width2  blurrness_devide
0    14.9300             16.916233                10
1    11.1750             14.946832                10
2    12.8525             14.520399                10
3    12.0450             15.094401                10
4    12.0750             14.276259                10
5    15.3700             16.706814                10
6    12.6500             14.447700                10
7     8.5875             16.872830                10
8    10.7375             16.528863                10
9    11.2950             14.969618                10
```

```
[ ]: import matplotlib.pyplot as plt
     import numpy as np


     labels = ['G1', 'G2', 'G3', 'G4', 'G5']
     men_means1 = [20, 34, 30, 35, 27]
     men_means2 = [20, 34, 30, 35, 27]
     men_means3 = [20, 34, 30, 35, 27]
     men_means4 = [20, 34, 30, 35, 27]
     men_means5 = [25, 32, 34, 20, 25]
     men_means6 = [25, 32, 34, 20, 25]

     x = np.arange(len(labels))  # the label locations
     width = 0.35  # the width of the bars

     fig, ax = plt.subplots()
     rects1 = ax.bar(x - width/6, men_means1, width, label='compression_level1')
     rects2 = ax.bar(x - width/6, men_means2, width, label='compression_level2')
     rects3 = ax.bar(x - width/6, men_means3, width, label='compression_level3')
     rects4 = ax.bar(x + width/6, men_means4, width, label='compression_level4')
     rects5 = ax.bar(x + width/6, men_means5, width, label='compression_level5')
     rects6 = ax.bar(x + width/6, men_means6, width, label='compression_level6')

     # Add some text for labels, title and custom x-axis tick labels, etc.
     ax.set_ylabel('Scores')
     ax.set_title('Scores by group and gender')
     ax.set_xticks(x, labels)
     ax.legend()

     ax.bar_label(rects1, padding=3)
     ax.bar_label(rects2, padding=3)
     ax.bar_label(rects3, padding=3)
     ax.bar_label(rects4, padding=3)
     ax.bar_label(rects5, padding=3)
     ax.bar_label(rects6, padding=3)

     fig.tight_layout()

     plt.show()
```

[137]:
```
features
```

[137]:
```
[                                                 image dominant_color  devide  \
 0   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
 1   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
 2   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
 3   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
 4   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
 ..                                               …            …     …
 78  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
 79  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
 80  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10
 81  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
 82  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10

     blurrness2  blurrness2_devide  average_pixel_width_devide  \
 0     9.260924                 10                          10
 1     9.509223                 10                          10
 2     8.217522                 10                          10
 3     9.176203                 10                          10
 4     8.920611                 10                          10
 ..         …                 …                          …
 78    8.510210                 10                          10
 79    9.517826                 10                          10
 80    9.732118                 10                          10
 81    9.768071                 10                          10
 82    8.611605                 10                          10

     dullness_devide  dullness2  average_pixel_width2  blurrness_devide
 0                 2    14.9300             16.916233                10
 1                 2    11.1750             14.946832                10
 2                 2    12.8525             14.520399                10
```

```
3                 2    12.0450             15.094401                10
4                 2    12.0750             14.276259                10
..              …        …                   …               …
78                2    14.8125             16.422526                10
79                2    11.2375             14.947917                10
80                2     5.8675             16.774089                10
81                2     8.0825             17.262370                10
82                2    10.7325             14.703776                10

[83 rows x 10 columns],
                                                image dominant_color  devide  \
0   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
1   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
2   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
3   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
4   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
..                                                …            …     …
79  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
80  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 17, 32]      10
81  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10
82  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10
83  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10

    blurrness2  blurrness2_devide  average_pixel_width_devide  \
0     8.966775                 10                          10
1     8.825283                 10                          10
2     7.637077                 10                          10
3     8.698621                 10                          10
4     8.314515                 10                          10
..         …                …                         …
79    8.007677                 10                          10
80    9.196445                 10                          10
81    9.117703                 10                          10
82    9.003062                 10                          10
83    7.755428                 10                          10

    dullness_devide  dullness2  average_pixel_width2  blurrness_devide
0                 2    14.0450             16.298828                10
1                 2    10.4750             15.002170                10
2                 2    11.1975             14.436849                10
3                 2    11.4125             14.616970                10
4                 2    11.8725             14.391276                10
..              …        …                   …               …
79                2    14.7775             16.019965                10
80                2    10.6575             14.739583                10
81                2     6.5175             16.759983                10
82                2     7.2025             17.184245                10
```

```
83                2    11.0275             14.389106                10

[84 rows x 10 columns],
                                                 image dominant_color  devide  \
0   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
1   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
2   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
3   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
4   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
..                                                …             …     …
79  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10
80  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
81  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 33]      10
82  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10
83  /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 30]      10

    blurrness2  blurrness2_devide  average_pixel_width_devide  \
0     7.571459                 10                          10
1     7.077818                 10                          10
2     6.292977                 10                          10
3     6.984635                 10                          10
4     6.947804                 10                          10
..         …                 …                          …
79    6.823091                 10                          10
80    7.576802                 10                          10
81    7.074446                 10                          10
82    7.162430                 10                          10
83    6.340002                 10                          10

    dullness_devide  dullness2  average_pixel_width2  blurrness_devide
0                 2    14.3725             16.303168                10
1                 2    11.8150             14.853516                10
2                 2    11.5525             14.618056                10
3                 2    13.2800             14.156901                10
4                 2    11.8950             14.367405                10
..              …        …                 …               …
79                2    16.3800             15.953776                10
80                2    11.4575             14.727648                10
81                2     7.6350             16.699219                10
82                2     8.0150             16.925998                10
83                2    12.9350             14.527995                10

[84 rows x 10 columns],
                                                 image dominant_color  devide  \
0   /home/jupyter/resources/out/video1/samle_pngs_…   [15, 16, 31]      10
1   /home/jupyter/resources/out/video1/samle_pngs_…   [16, 16, 32]      10
2   /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 31]      10
```

```
3   /home/jupyter/resources/out/video1/samle_pngs_…   [16, 16, 32]      10
4   /home/jupyter/resources/out/video1/samle_pngs_…   [16, 16, 32]      10
..                                                …            …    …
79  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 16, 31]      10
80  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 16, 32]      10
81  /home/jupyter/resources/out/video1/samle_pngs_…   [17, 17, 34]      10
82  /home/jupyter/resources/out/video1/samle_pngs_…   [17, 17, 33]      10
83  /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 31]      10

    blurrness2  blurrness2_devide  average_pixel_width_devide  \
0     5.406805                 10                          10
1     4.464937                 10                          10
2     4.333433                 10                          10
3     4.286006                 10                          10
4     4.086361                 10                          10
..         …                …                          …
79    5.191774                 10                          10
80    4.785688                 10                          10
81    4.035429                 10                          10
82    4.002675                 10                          10
83    4.416225                 10                          10

    dullness_devide  dullness2  average_pixel_width2  blurrness_devide
0                 2    14.9325             14.851345                10
1                 2    13.5900             13.672960                10
2                 2    10.6900             13.255208                10
3                 2    13.5675             13.496094                10
4                 2    11.8575             12.601997                10
..              …        …                   …               …
79                2    15.6025             14.922960                10
80                2    12.7300             13.868273                10
81                2     6.3050             14.975043                10
82                2     8.2200             15.248481                10
83                2    10.6600             13.146701                10

[84 rows x 10 columns],
                                                image dominant_color  devide  \
0   /home/jupyter/resources/out/video1/samle_pngs_…   [16, 17, 32]      10
1   /home/jupyter/resources/out/video1/samle_pngs_…   [18, 17, 33]      10
2   /home/jupyter/resources/out/video1/samle_pngs_…   [18, 17, 33]      10
3   /home/jupyter/resources/out/video1/samle_pngs_…   [18, 17, 33]      10
4   /home/jupyter/resources/out/video1/samle_pngs_…   [18, 17, 33]      10
..                                                …            …    …
79  /home/jupyter/resources/out/video1/samle_pngs_…   [46, 45, 73]      10
80  /home/jupyter/resources/out/video1/samle_pngs_…   [17, 17, 33]      10
81  /home/jupyter/resources/out/video1/samle_pngs_…   [19, 19, 36]      10
82  /home/jupyter/resources/out/video1/samle_pngs_…   [19, 19, 36]      10
```

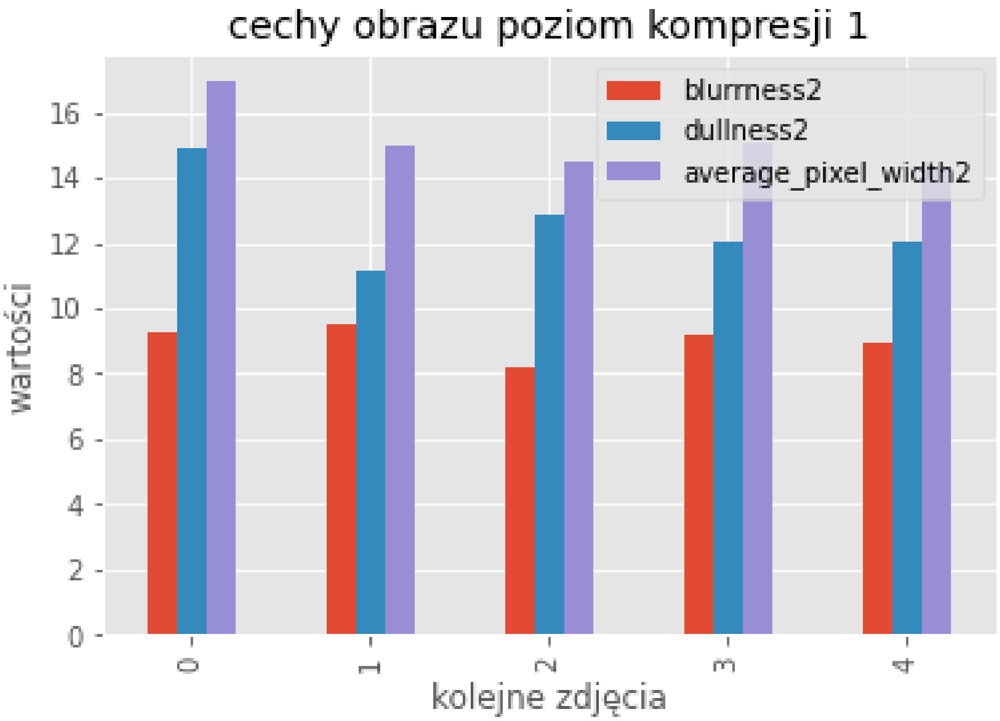
```
83  /home/jupyter/resources/out/video1/samle_pngs_…  [18, 17, 33]      10

    blurrness2  blurrness2_devide  average_pixel_width_devide  \
0     3.340039                 10                          10
1     2.058169                 10                          10
2     2.406318                 10                          10
3     2.171787                 10                          10
4     2.084418                 10                          10
..         …                 …                          …
79    2.940397                 10                          10
80    2.359128                 10                          10
81    2.114911                 10                          10
82    1.958340                 10                          10
83    2.388421                 10                          10

    dullness_devide  dullness2  average_pixel_width2  blurrness_devide
0                 2    15.5275             11.804470                10
1                 2    15.9525             10.130208                10
2                 2    17.2350             10.851780                10
3                 2    16.1675             10.333116                10
4                 2    16.1825              9.720052                10
..              …        …                  …               …
79                2    15.4675             11.816406                10
80                2    16.0950             10.593533                10
81                2    10.0000             11.475694                10
82                2     9.3925             11.315104                10
83                2    17.1450             10.817057                10

[84 rows x 10 columns],
                                               image dominant_color  devide  \
0   /home/jupyter/resources/out/video1/samle_pngs_…  [17, 18, 35]      10
1   /home/jupyter/resources/out/video1/samle_pngs_…  [18, 18, 33]      10
2   /home/jupyter/resources/out/video1/samle_pngs_…  [18, 18, 33]      10
3   /home/jupyter/resources/out/video1/samle_pngs_…  [17, 18, 33]      10
4   /home/jupyter/resources/out/video1/samle_pngs_…  [18, 18, 33]      10
..                                                …             …     …
79  /home/jupyter/resources/out/video1/samle_pngs_…  [17, 18, 35]      10
80  /home/jupyter/resources/out/video1/samle_pngs_…  [17, 18, 33]      10
81  /home/jupyter/resources/out/video1/samle_pngs_…  [20, 20, 37]      10
82  /home/jupyter/resources/out/video1/samle_pngs_…  [20, 20, 37]      10
83  /home/jupyter/resources/out/video1/samle_pngs_…  [18, 18, 33]      10

    blurrness2  blurrness2_devide  average_pixel_width_devide  \
0     3.297118                 10                          10
1     1.950436                 10                          10
2     2.244680                 10                          10
3     2.008312                 10                          10
```

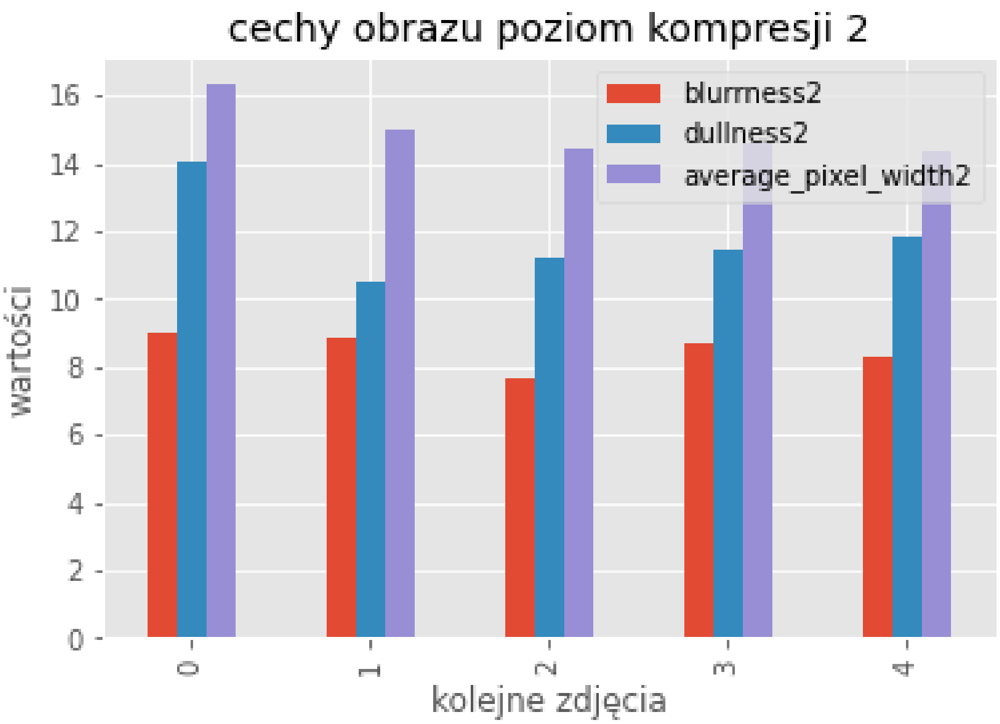
```
4     1.770200                 10                            10
..         …                …                          …
79    2.924378                 10                            10
80    2.155240                 10                            10
81    1.883468                 10                            10
82    1.817045                 10                            10
83    2.286363                 10                            10

    dullness_devide  dullness2  average_pixel_width2  blurrness_devide
0                  2    16.0600             11.735026                10
1                  2    15.4650              9.786241                10
2                  2    15.4925             10.793186                10
3                  2    14.9225              9.597439                10
4                  2    15.0450              9.888238                10
..              …        …                    …               …
79                 2    16.0750             11.881510                10
80                 2    15.5200             10.003255                10
81                 2     6.6775             11.103516                10
82                 2     8.6950             11.001519                10
83                 2    15.4550             10.629340                10

[84 rows x 10 columns]]
```

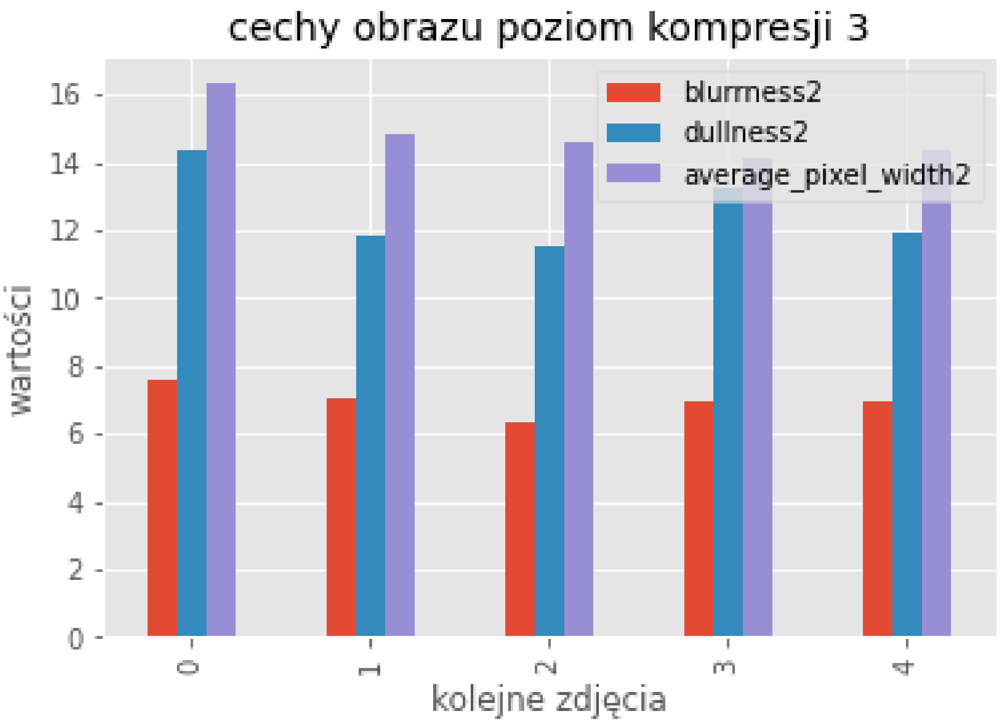
[153]:

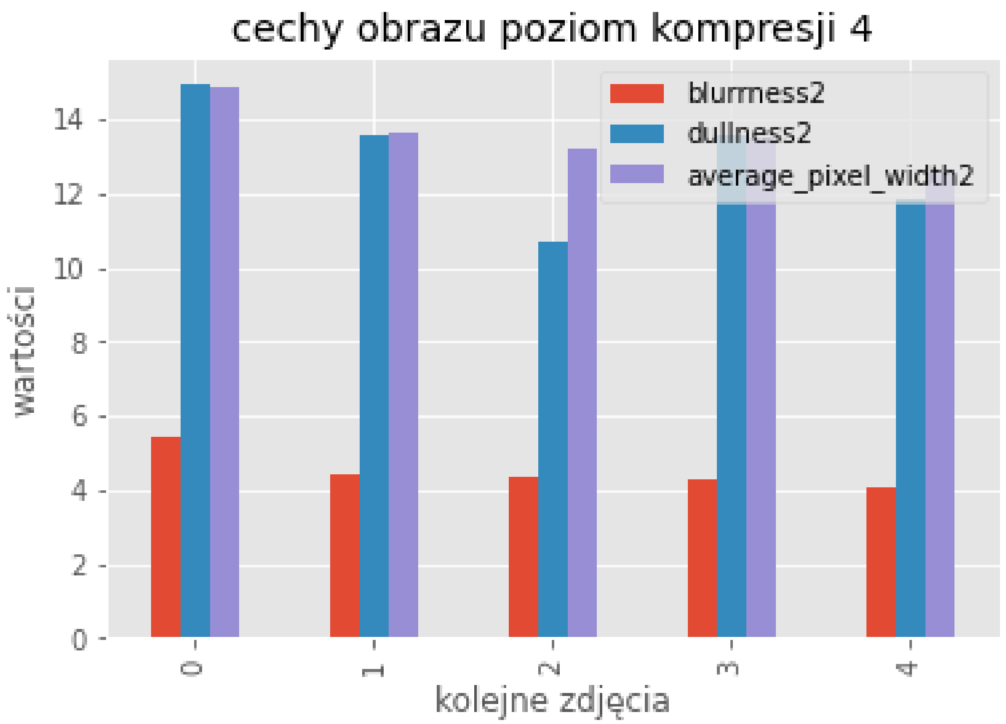
```
for i in range(0, 6):
    # features[i]["blurrness_devide"] = 10
    # features[i]["average_pixel_width_devide"] = 10
    # features[i]["dullness_devide"] = 2
    pass

for i in range(0, 6):
    # features[i]["dullness2"] = features[i]["dullness"] /␣
 ↪features[i]["dullness_devide"]
    # features[i]["average_pixel_width2"] = features[i]["average_pixel_width"]␣
 ↪* features[i]["average_pixel_width_devide"]
    # features[i]["blurrness2"] = features[i]["blurrness"] /␣
 ↪features[i]["blurrness_devide"]

    # del features[i]["dullness"]
    # del features[i]["average_pixel_width"]
    # del features[i]["blurrness"]
    # del features[i]["devide"]
    # del features[i]["blurrness_devide"]
    # del features[i]["average_pixel_width_devide"]
    # del features[i]["dullness_devide"]
    pass

for i in range(0, 6):
```

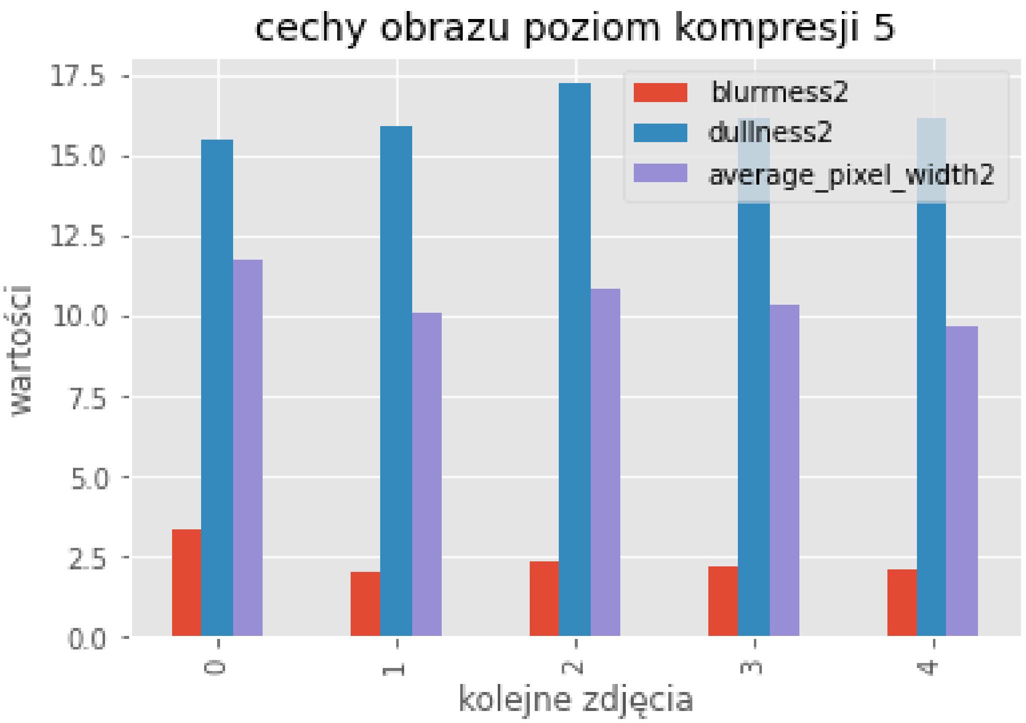
```
features[i].head(5).plot(kind="bar")

plt.title(f"cechy obrazu poziom kompresji {i+1}")
plt.xlabel("kolejne zdjęcia")
plt.ylabel("wartości")
ax.legend(bbox_to_anchor=(1.1, 1.05))
```

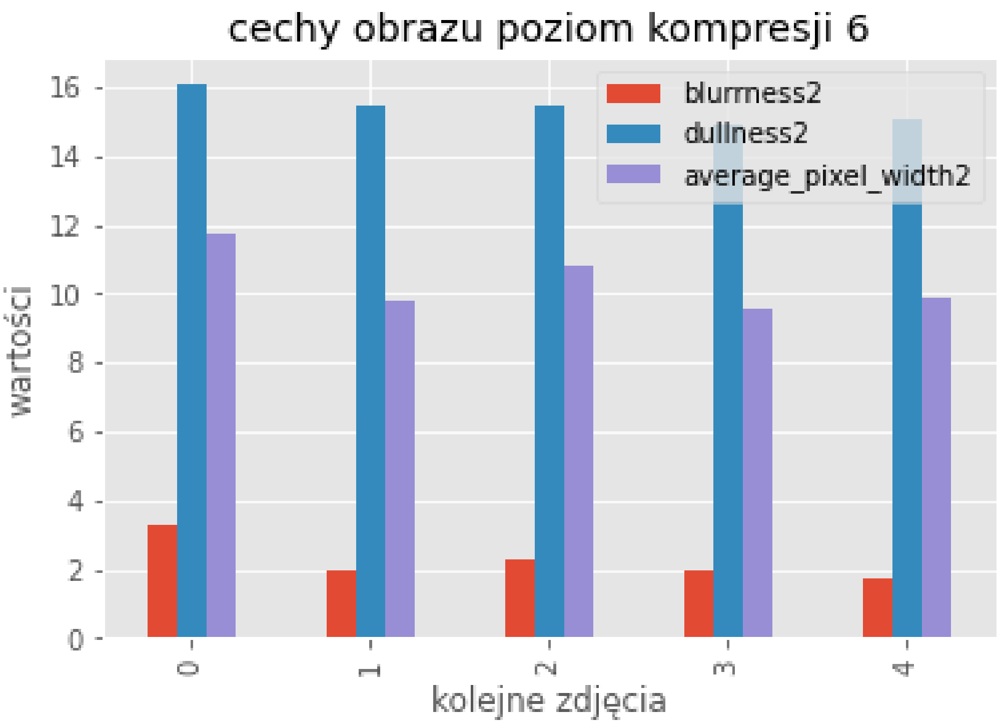



cechy obrazu poziom kompresji 2
blurrness2
dullness2
average_pixel_width2
wartości
kolejne zdjęcia
0
2
4
6
8
10
12
14
16
0
1
2
3
4


cechy obrazu poziom kompresji 3
blurrness2
dullness2
average_pixel_width2
wartości
kolejne zdjęcia
0
2
4
6
8
10
12
14
16
0
1
2
3
4


cechy obrazu poziom kompresji 4
blurrness2
dullness2
average_pixel_width2
wartości
0
2
4
6
8
10
12
14
0
1
2
3
4
kolejne zdjęcia


cechy obrazu poziom kompresji 5
blurrness2
dullness2
average_pixel_width2
wartości
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
1
2
3
4
kolejne zdjęcia



[ ]:

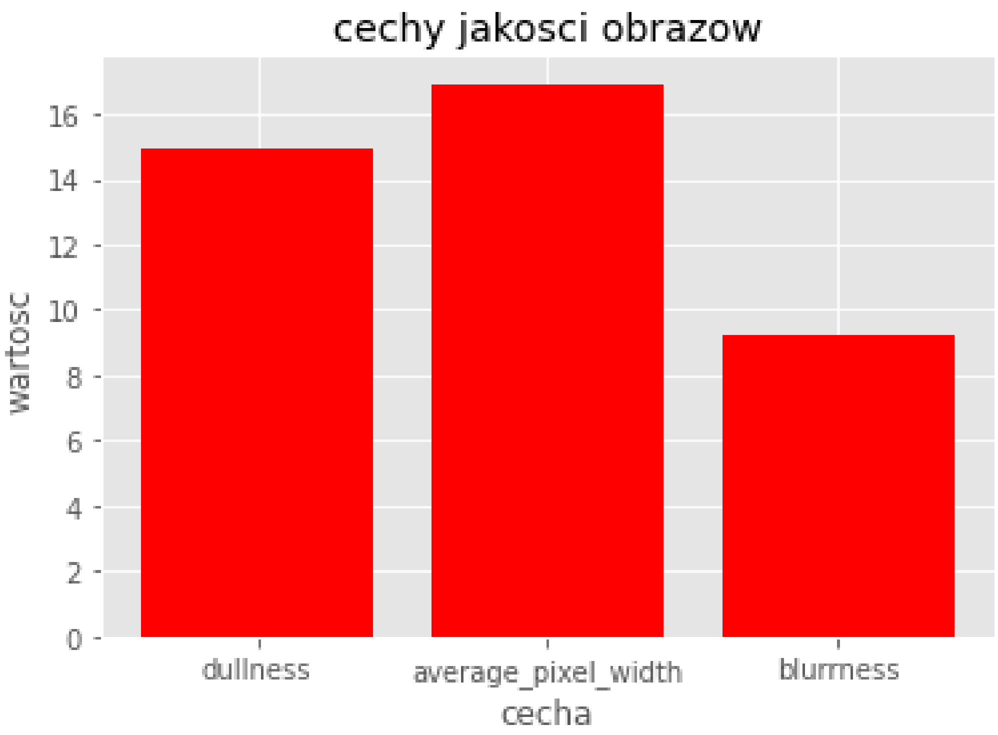
```
for j in range(0, NUMBER):
```

[64]:

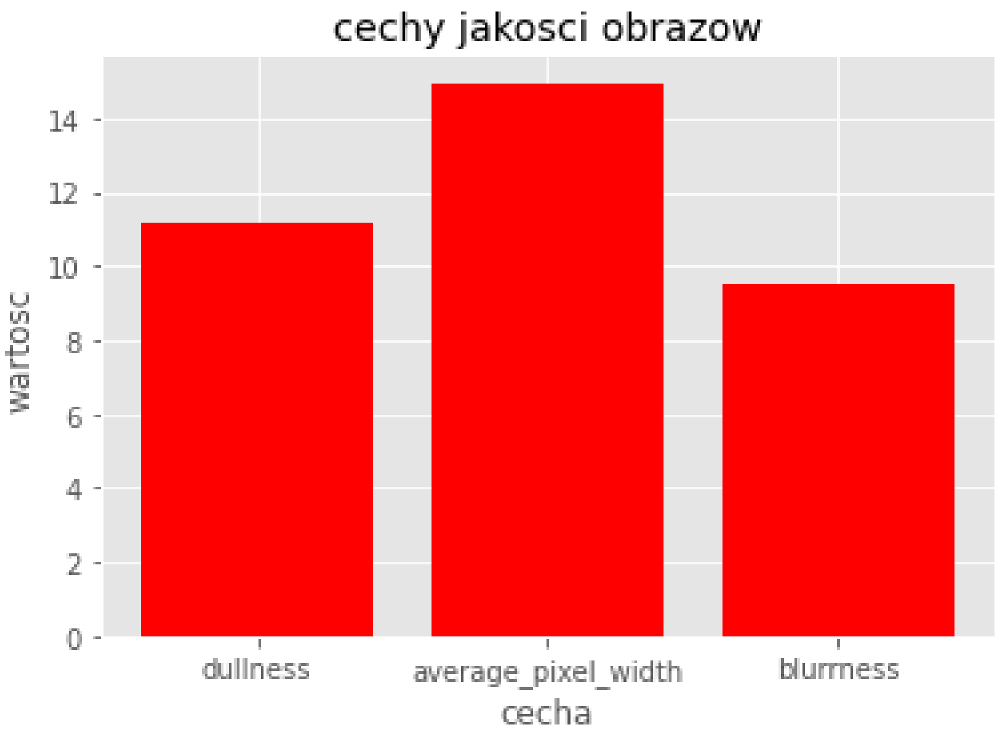
```
import matplotlib.pyplot as plt
import random
%matplotlib inline
plt.style.use('ggplot')

NUMBER=4

# x = ['dullness', 'whiteness', 'average_pixel_width', 'blurrness']
x = ['dullness', 'average_pixel_width', 'blurrness']

def show_chart(lables, values, col="green"):
    plt.bar(x_pos, values, color=f"{col}")
    plt.xlabel("cecha")
    plt.ylabel("wartosc")
    plt.title("cechy jakosci obrazow")
    plt.xticks(x_pos, x)
    plt.show()

# features["dullness"]
# features['whiteness']
# features['dominant_color']
```

```python
# features['average_pixel_width']
# features['blurrness']

x_pos = [i for i, _ in enumerate(x)]
color = "blue"

colour = [ "red", "blue", "green", "yellow", "purple", "orange", "white", "black" ]

for i in range(0, 6):
    col = random.choice(colour)

    for j in range(0, NUMBER):
        print(f"compression level {i}")

        vals = [
            features[i]['dullness'][j] / 2,
            # features[i]['whiteness'][j],
            features[i]['average_pixel_width'][j] *10,
            features[i]["blurrness"][j] / 10,
        ]
        print(vals)
        show_chart(x_pos, vals, col)
```

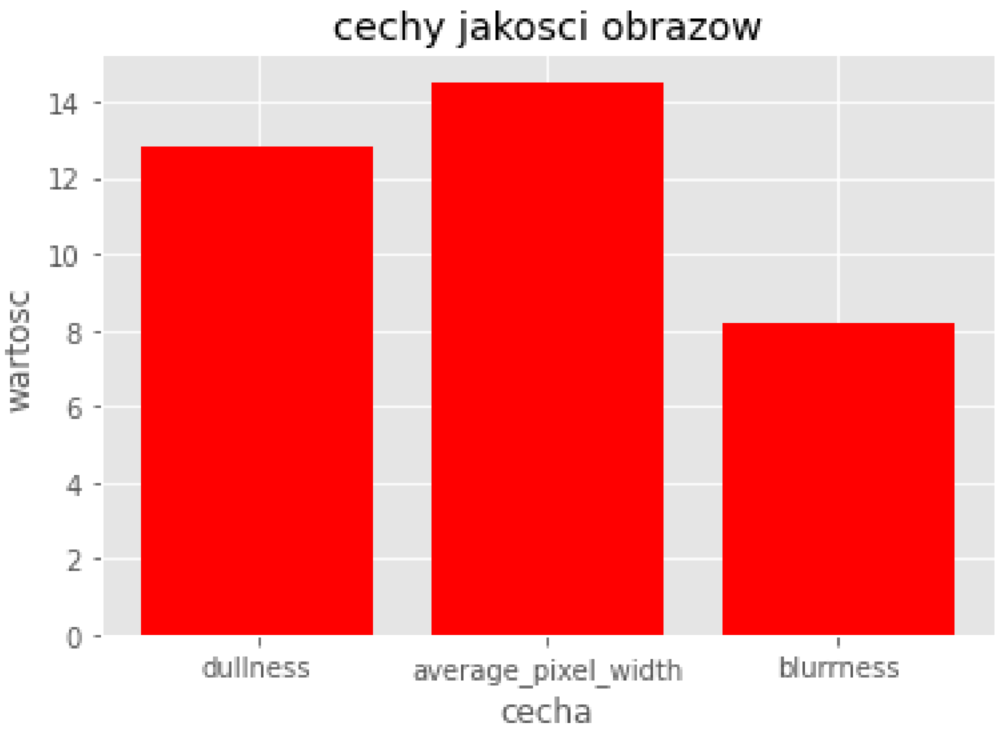
```
compression level 0
[14.93, 16.91623263888889, 9.260923693587575]
```



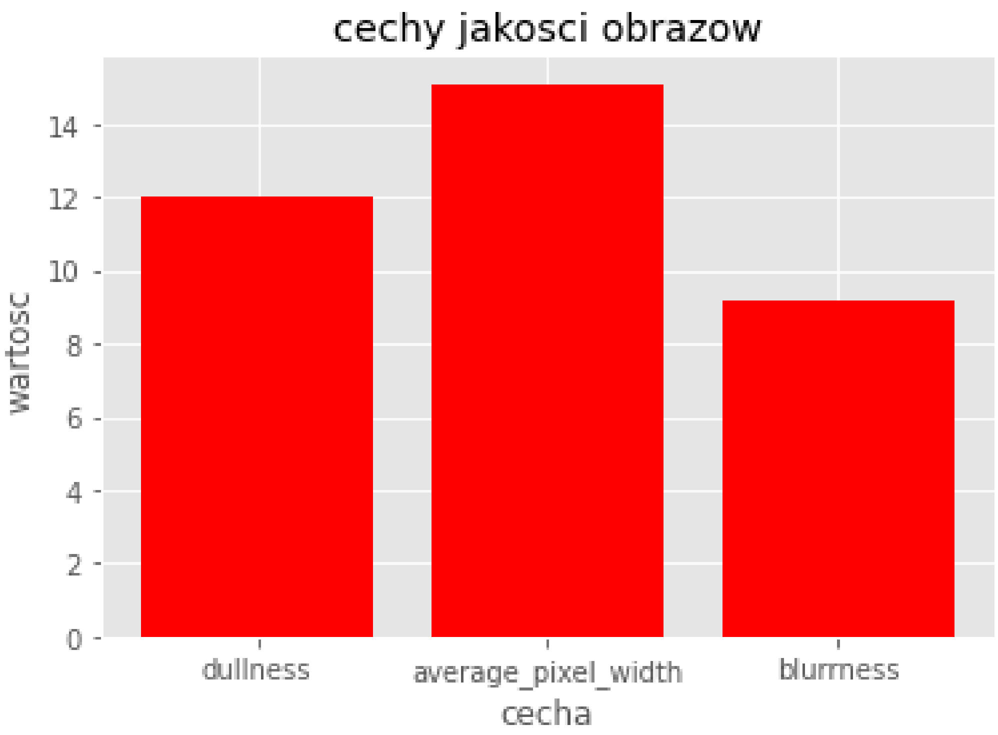
```
compression level 0
[11.175, 14.946831597222221, 9.509223044386498]
```



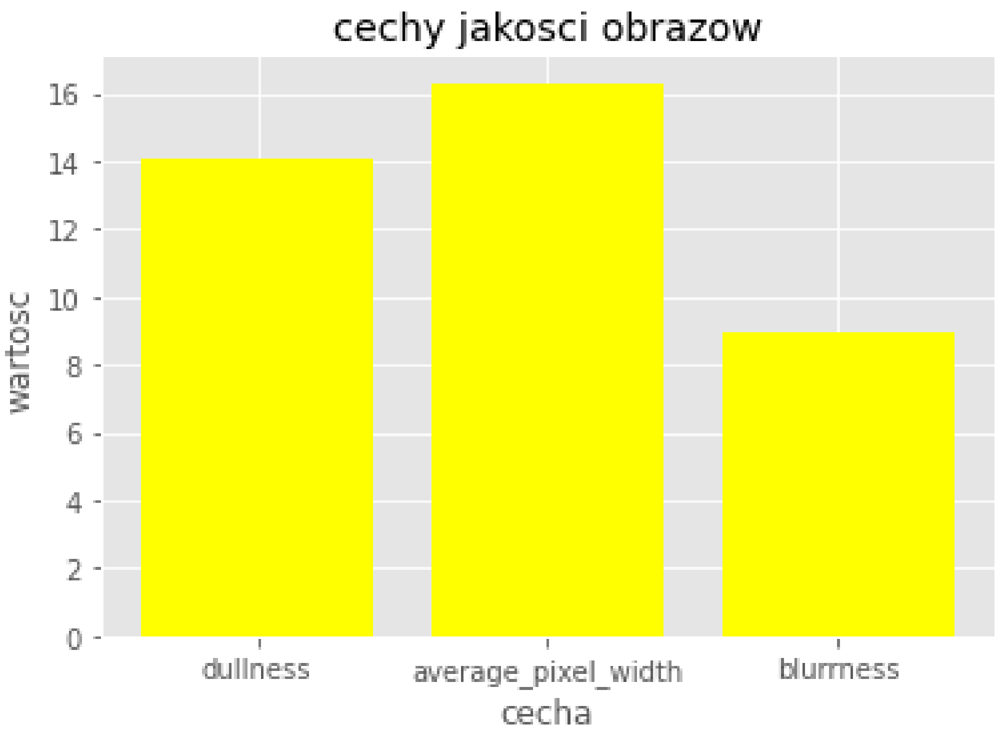
```
compression level 0
[12.852500000000001, 14.520399305555555, 8.217522306696456]
```



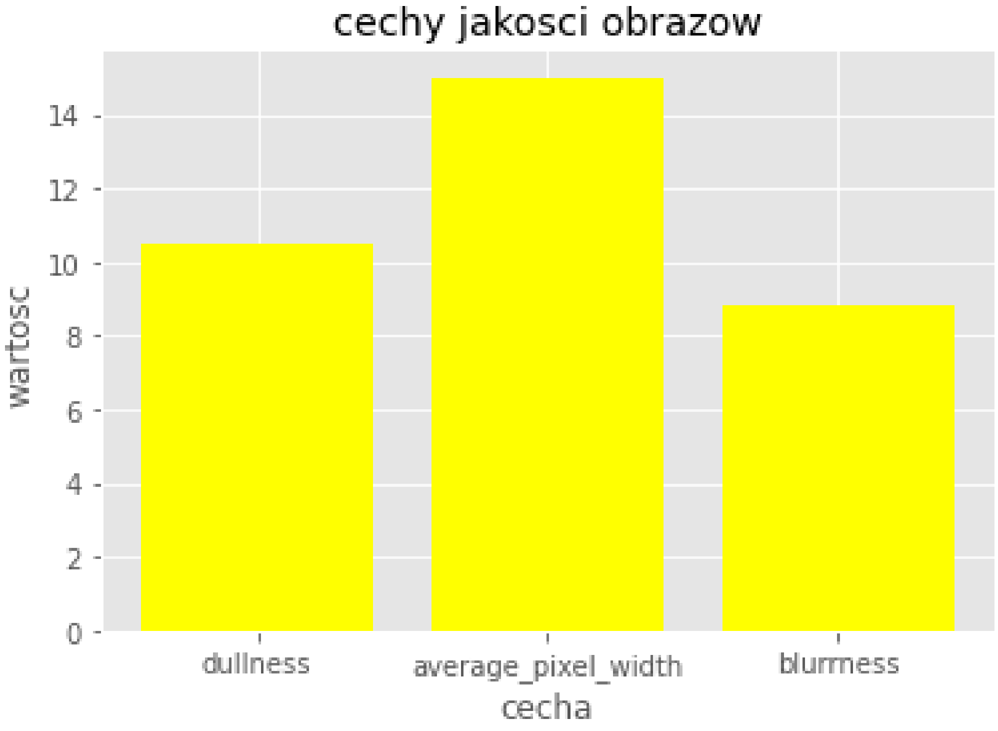
```
compression level 0
[12.045, 15.094401041666668, 9.17620279221523]
```



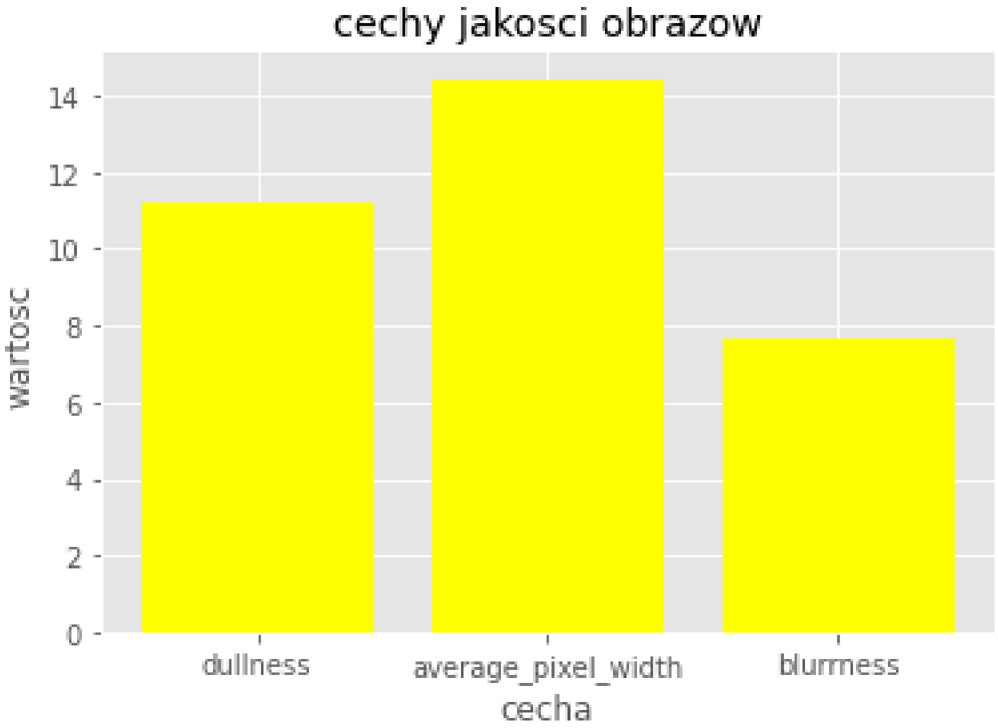
```
compression level 1
[14.045, 16.298828125000004, 8.966775168396632]
```



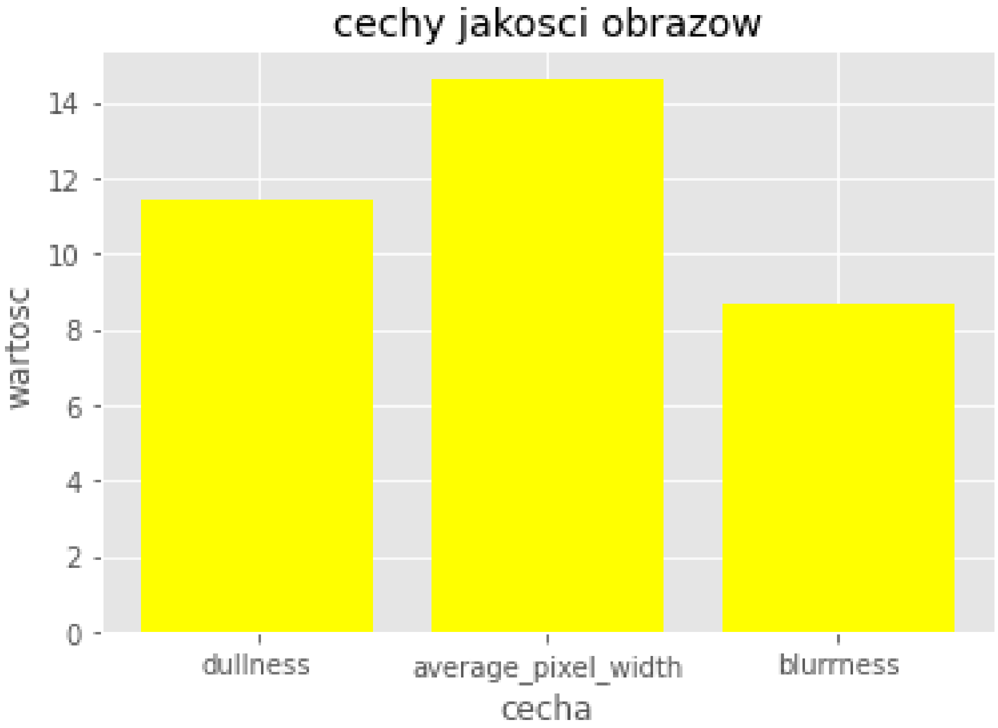
```
compression level 1
[10.475, 15.00217013888889, 8.825282631145994]
```



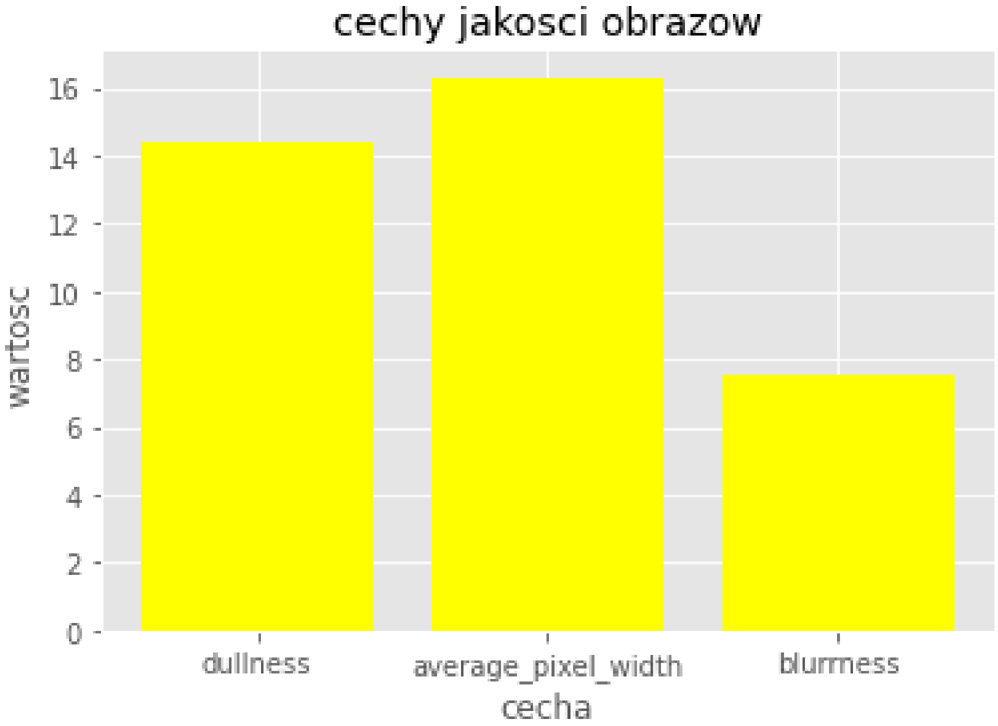
```
compression level 1
[11.1975, 14.436848958333332, 7.637076858767284]
```



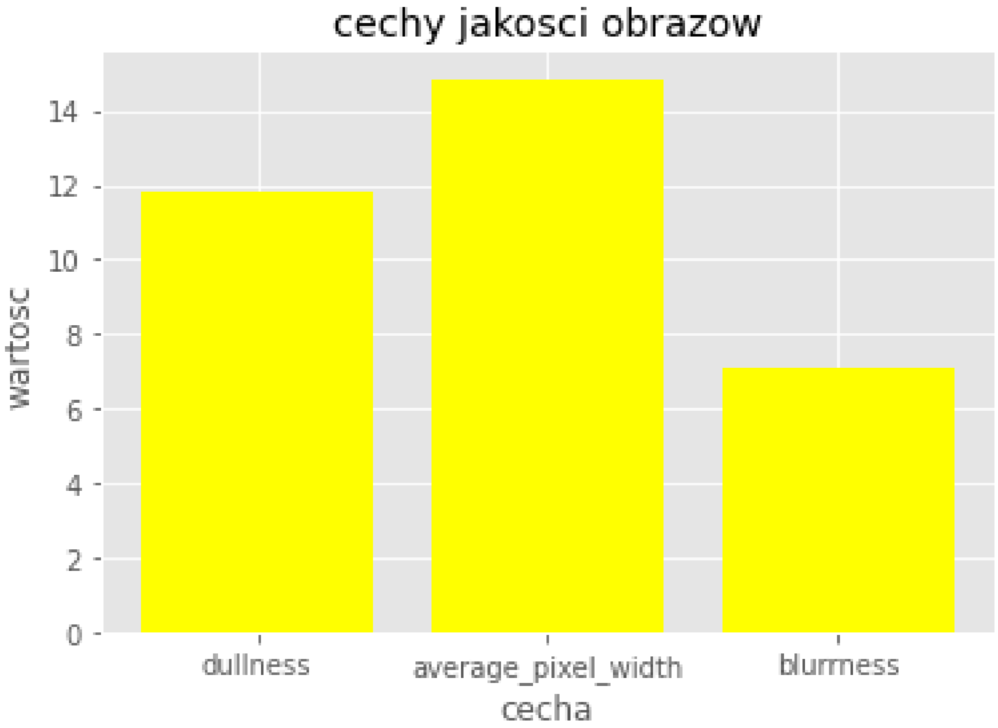
```
compression level 1
[11.4125, 14.61697048611111, 8.69862092882215]
```



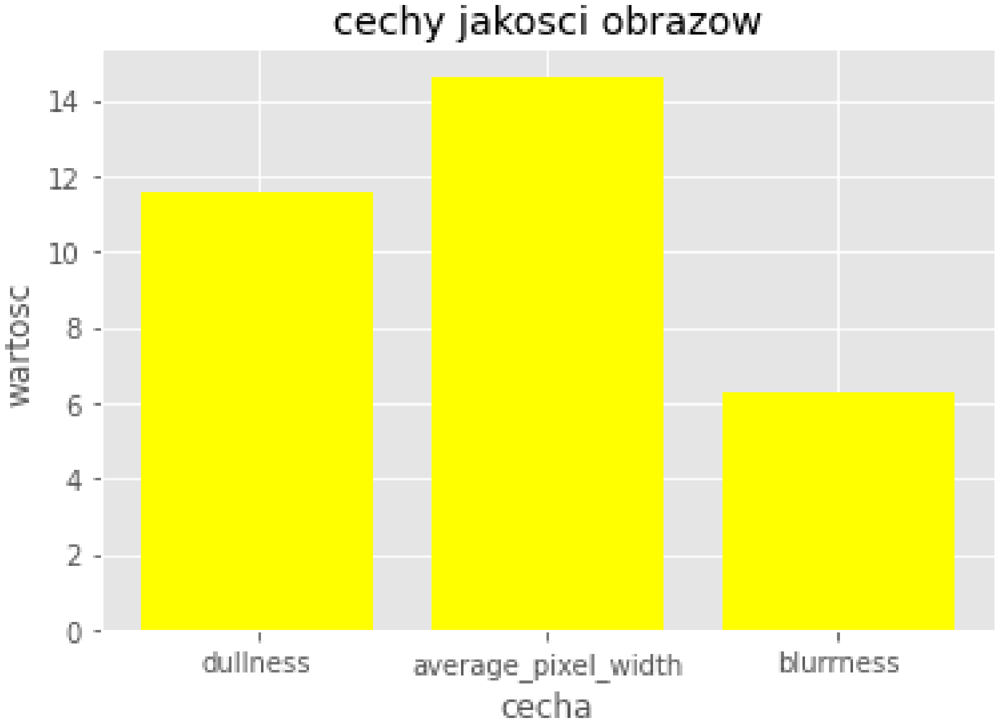
```
compression level 2
[14.3725, 16.30316840277778, 7.571459067035133]
```



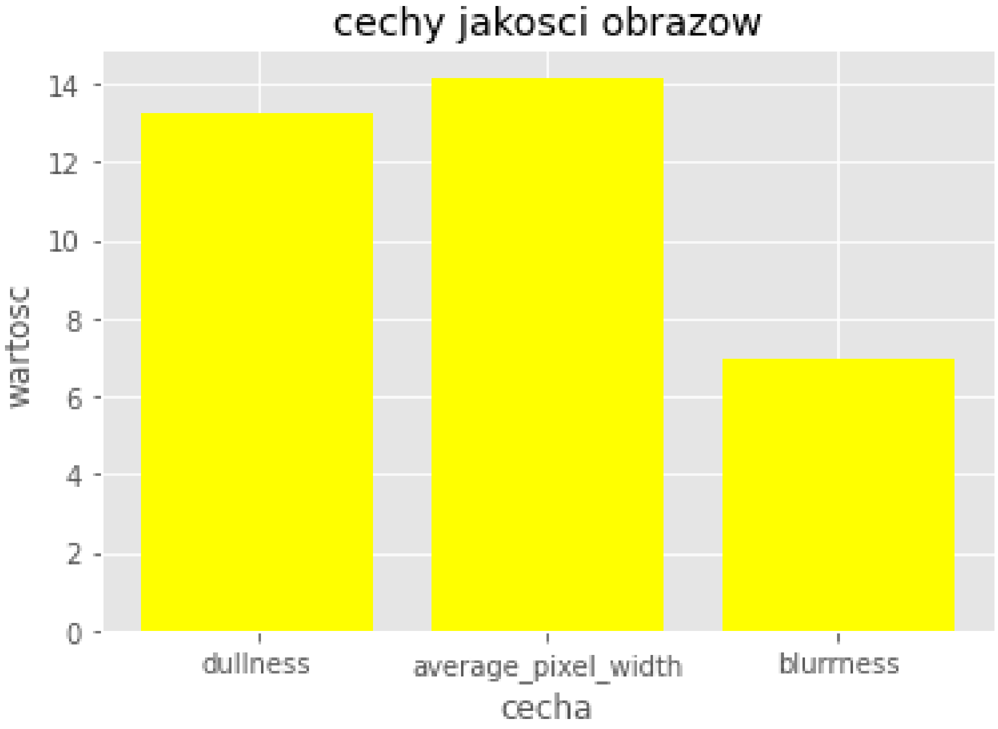
```
compression level 2
[11.815, 14.853515625, 7.077817686283325]
```



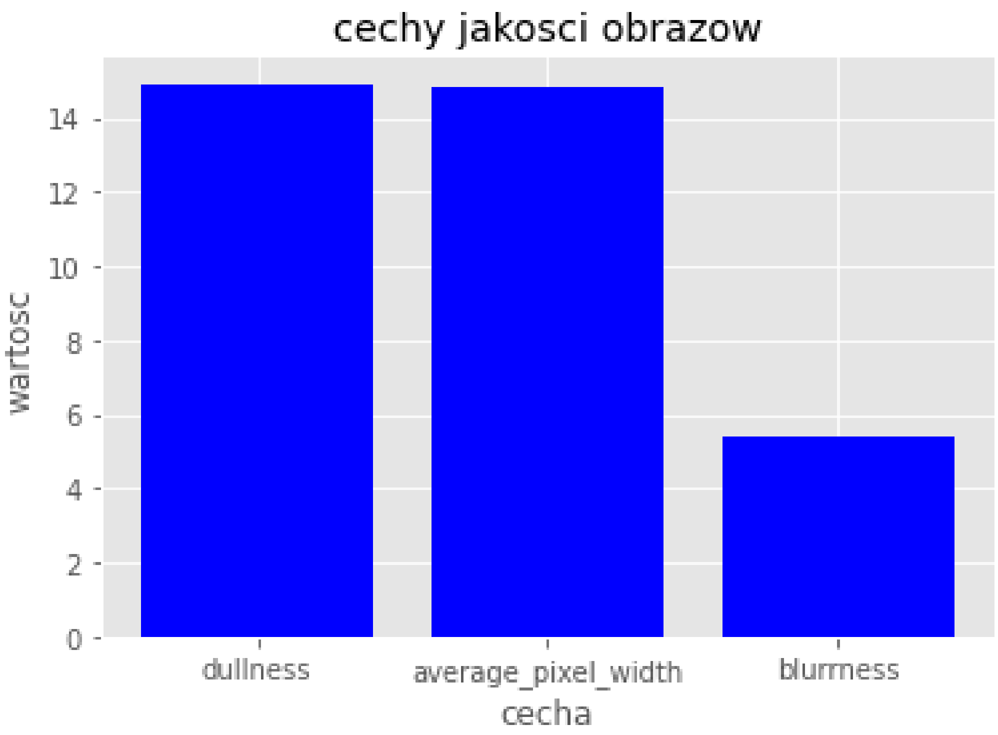
```
compression level 2
[11.5525, 14.618055555555555, 6.292976636755199]
```



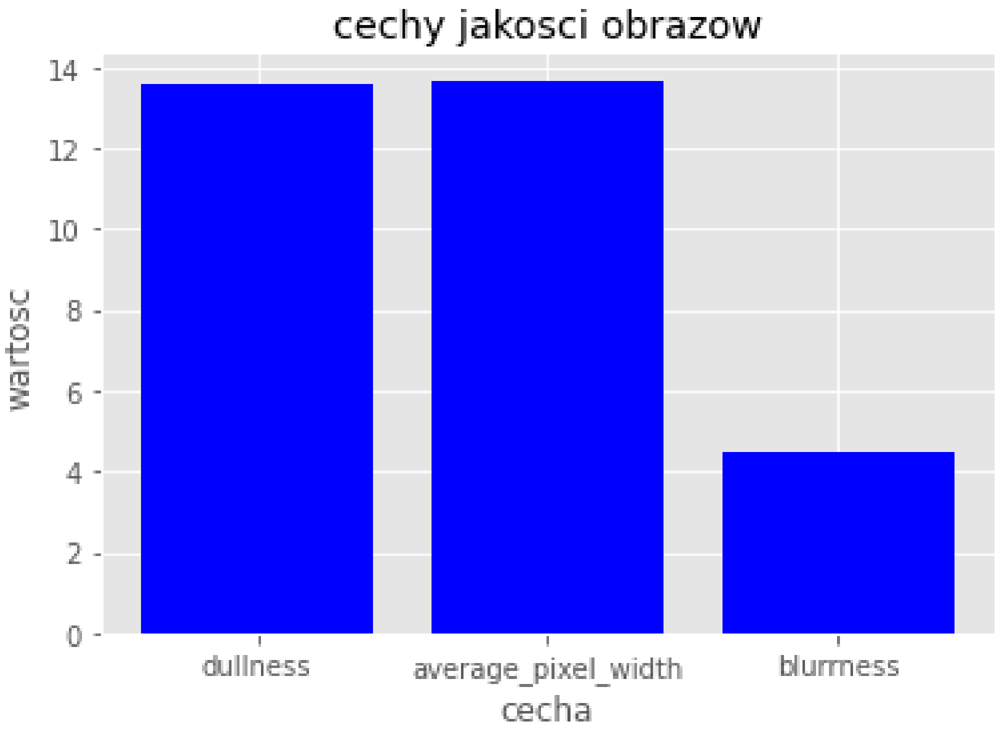
```
compression level 2
[13.280000000000001, 14.156901041666664, 6.984635482025027]
```



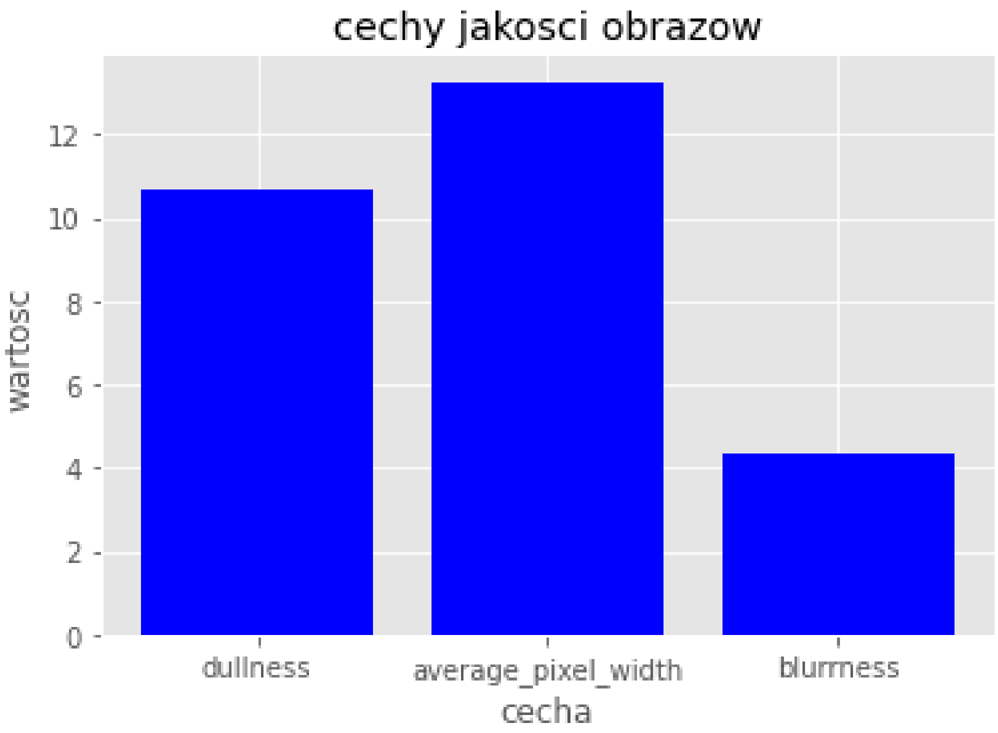
```
compression level 3
[14.9325, 14.85134548611111, 5.406804983302809]
```



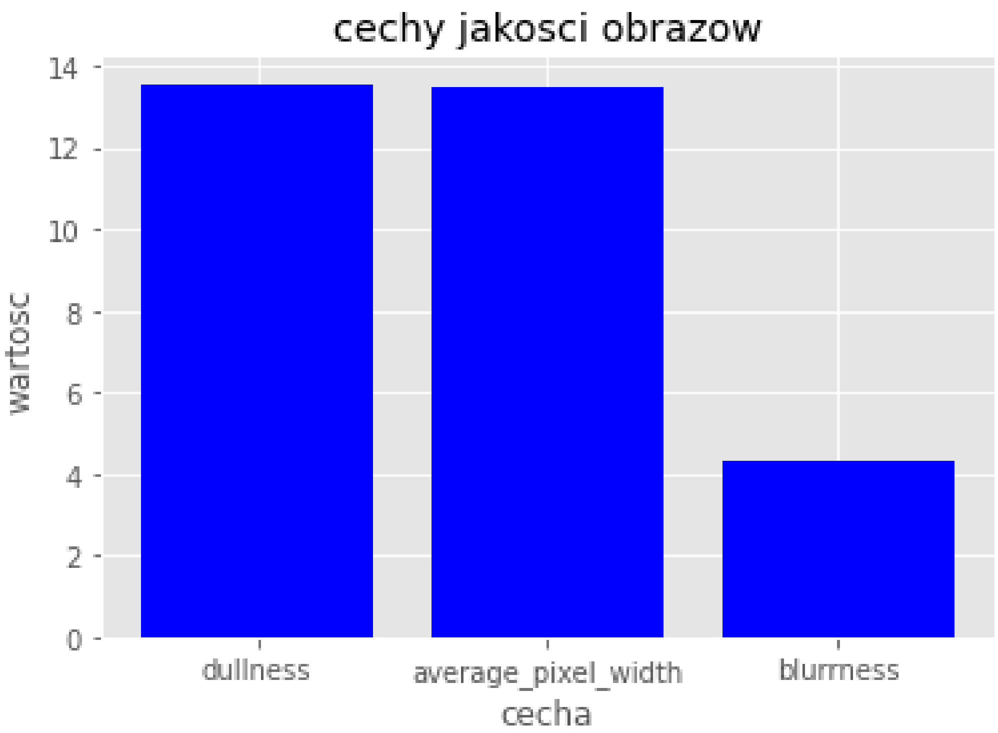
```
compression level 3
[13.59, 13.672960069444445, 4.464936756039725]
```



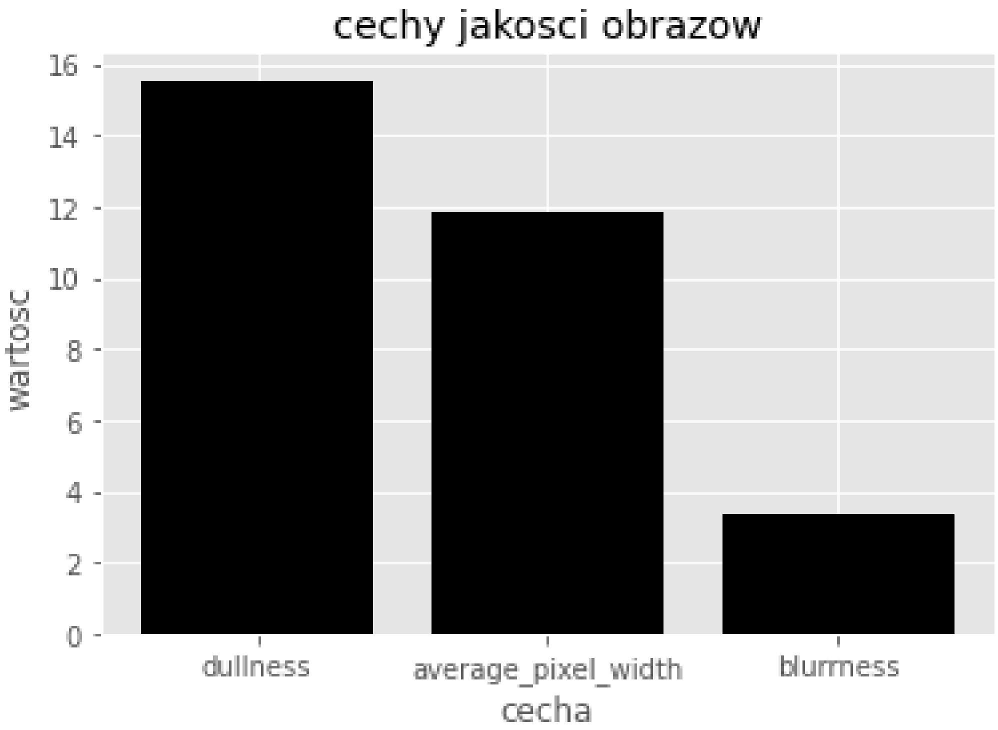
```
compression level 3
[10.69, 13.255208333333336, 4.3334329883988705]
```



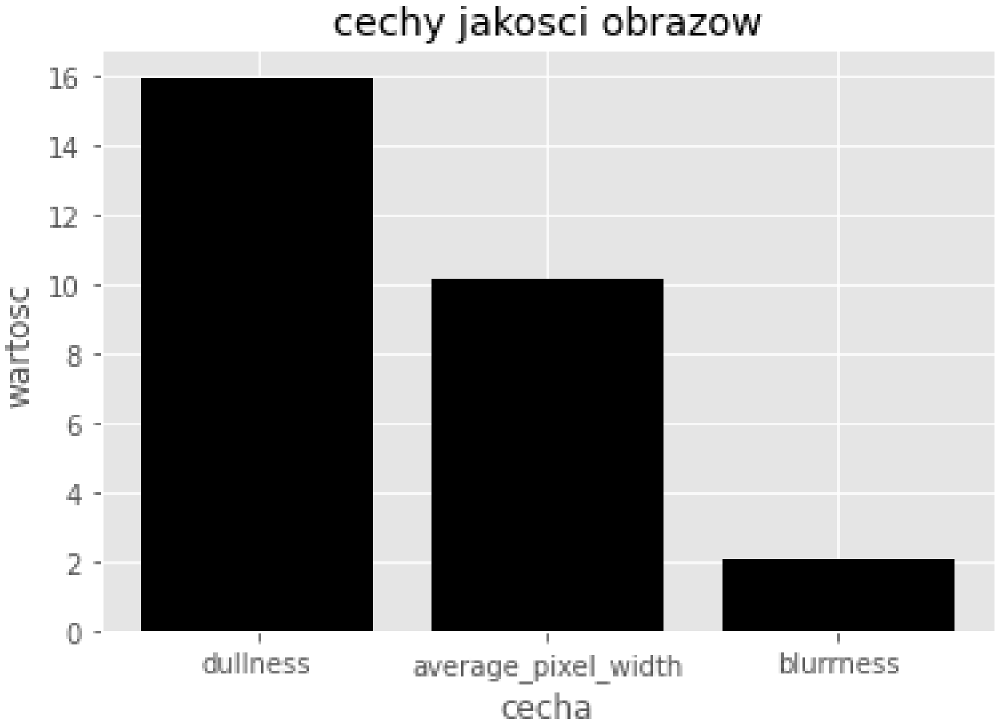
```
compression level 3
[13.567499999999999, 13.49609375, 4.2860057486073755]
```



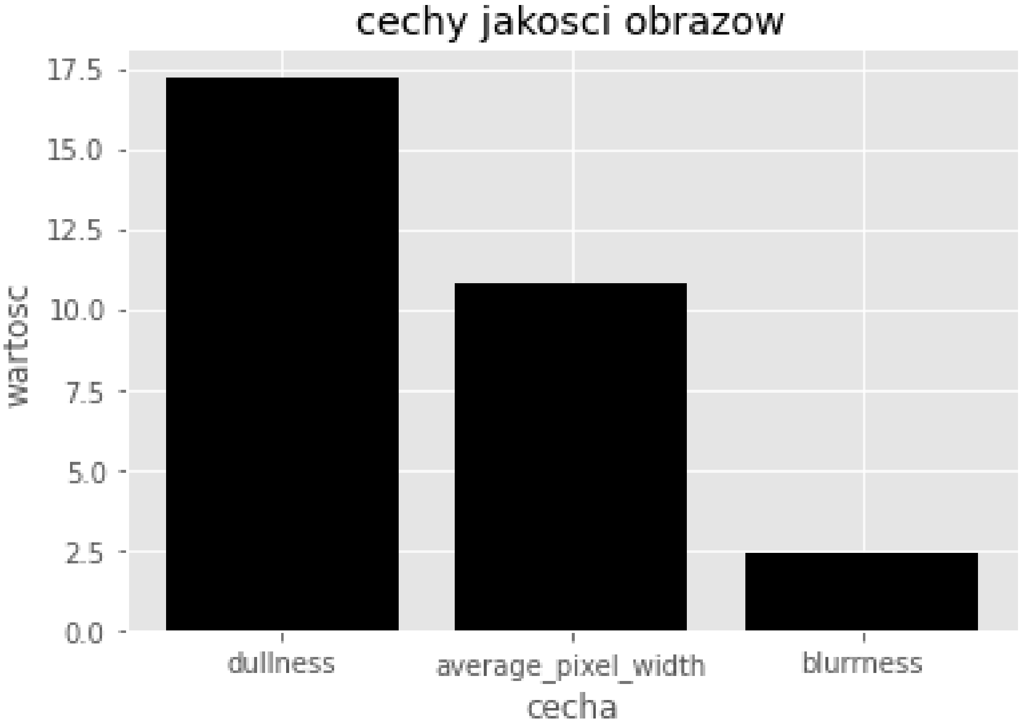
```
compression level 4
[15.5275, 11.80447048611111, 3.340038623846431]
```



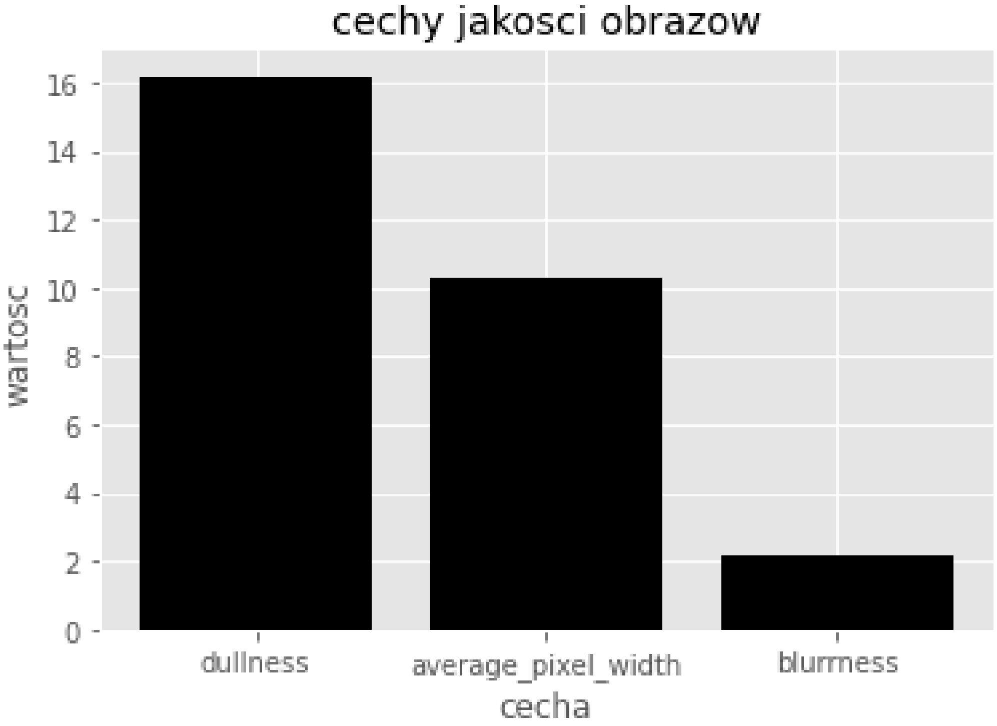
```
compression level 4
[15.9525, 10.130208333333332, 2.05816925639388]
```



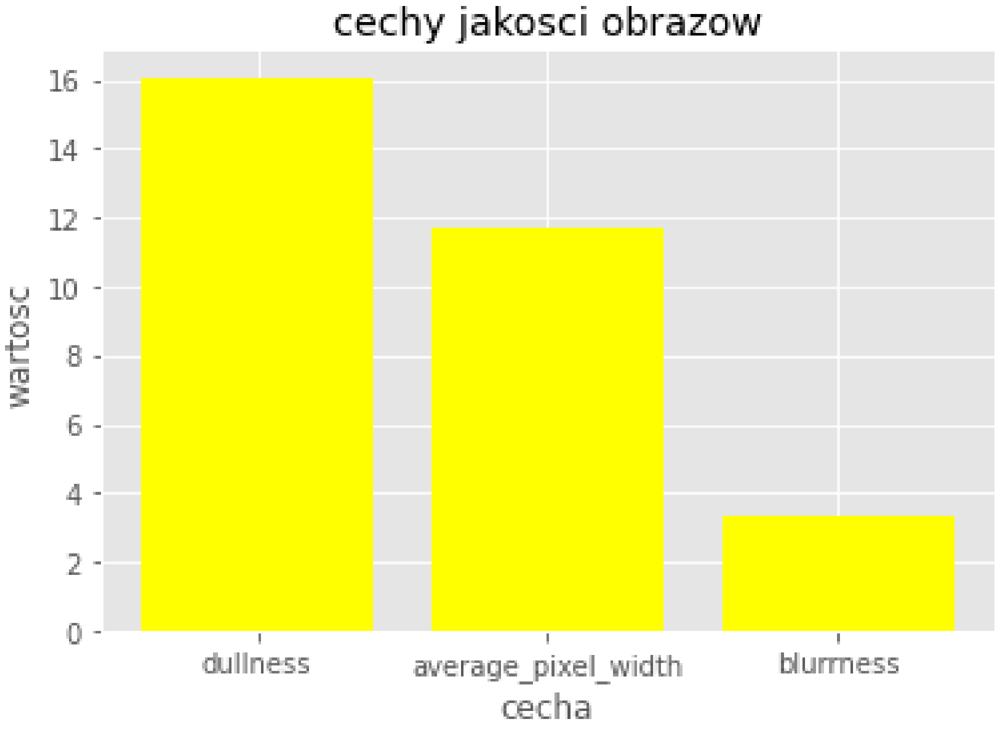
```
compression level 4
[17.235, 10.85177951388889, 2.4063183734733675]
```



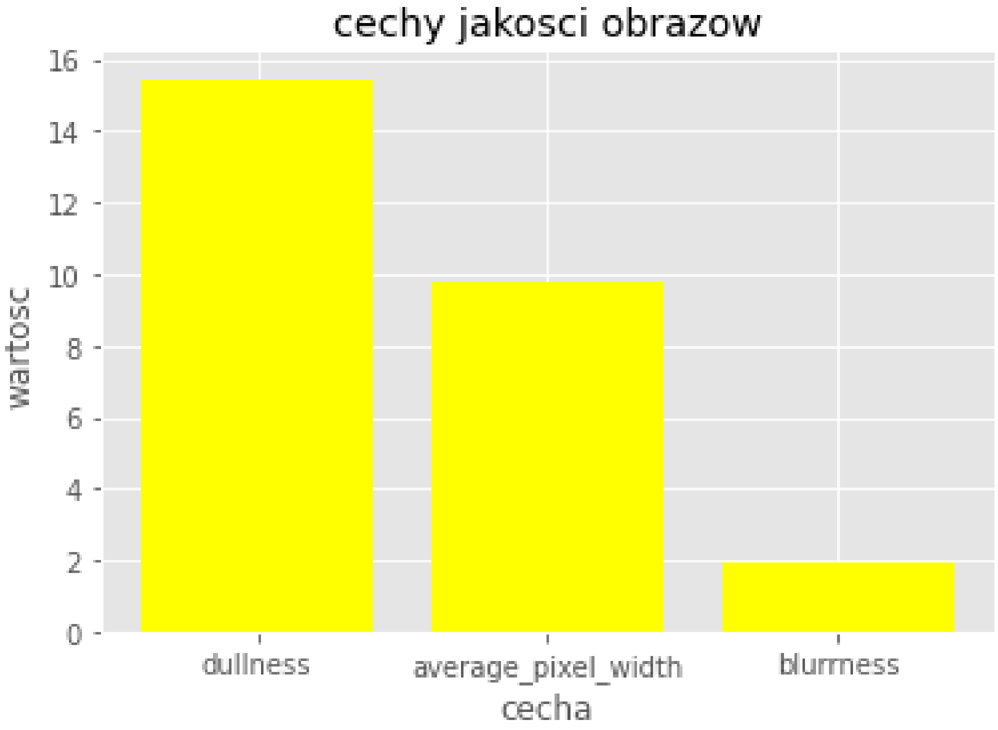
```
compression level 4
[16.1675, 10.333116319444445, 2.171786815275498]
```



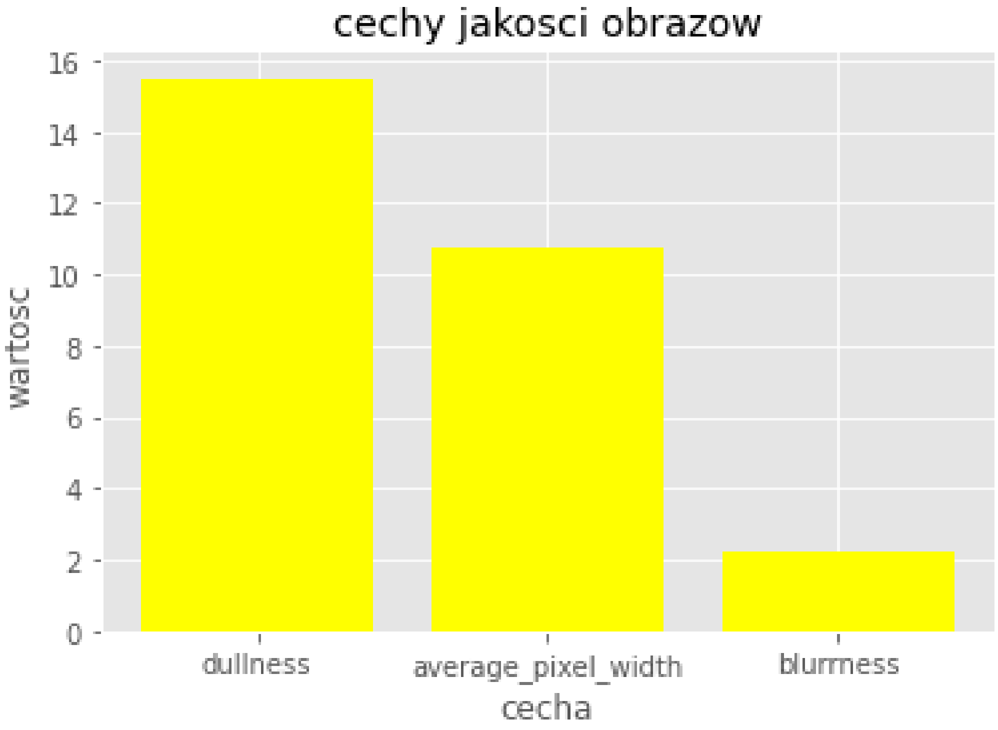
```
compression level 5
[16.06, 11.735026041666668, 3.297118381014106]
```



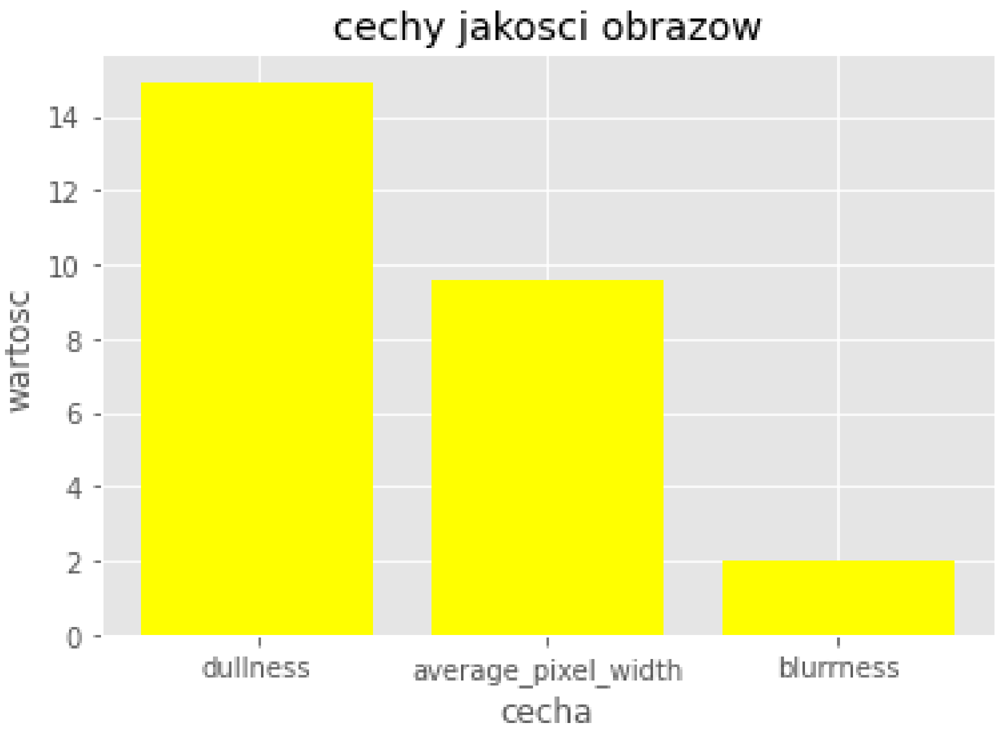
```
compression level 5
[15.465, 9.786241319444445, 1.9504362941784321]
```



```
compression level 5
[15.4925, 10.79318576388889, 2.2446800643015488]
```



```
compression level 5
[14.9225, 9.59743923611111, 2.008312396339016]
```



[41]:
```
features[1]
```

[41]:
```
                                                    image  dullness  whiteness  \
0   /home/jupyter/resources/out/video1/samle_pngs_…    28.090        0.0
1   /home/jupyter/resources/out/video1/samle_pngs_…    20.950        0.0
2   /home/jupyter/resources/out/video1/samle_pngs_…    22.395        0.0
3   /home/jupyter/resources/out/video1/samle_pngs_…    22.825        0.0
4   /home/jupyter/resources/out/video1/samle_pngs_…    23.745        0.0
..                                              …         …         …
79  /home/jupyter/resources/out/video1/samle_pngs_…    29.555        0.0
80  /home/jupyter/resources/out/video1/samle_pngs_…    21.315        0.0
81  /home/jupyter/resources/out/video1/samle_pngs_…    13.035        0.0
82  /home/jupyter/resources/out/video1/samle_pngs_…    14.405        0.0
83  /home/jupyter/resources/out/video1/samle_pngs_…    22.055        0.0

    average_pixel_width dominant_color  blurrness
0              1.629883   [15, 16, 30]  89.667752
1              1.500217   [15, 17, 32]  88.252826
```

```
2             1.443685   [15, 16, 30]  76.370769
3             1.461697   [15, 16, 31]  86.986209
4             1.439128   [15, 16, 31]  83.145146
..                 …            …          …
79            1.601997   [15, 16, 30]  80.076765
80            1.473958   [15, 17, 32]  91.964455
81            1.675998   [16, 17, 32]  91.177032
82            1.718424   [16, 17, 32]  90.030621
83            1.438911   [15, 16, 30]  77.554279

[84 rows x 6 columns]
```

[31]:

```
from imageai.Classification import ImageClassification
import os

execution_path = os.getcwd()

prediction = ImageClassification()
prediction.setModelTypeAsResNet50()
prediction.setModelPath(os.path.join("/home/jupyter/resources/
 ↪resnet50_imagenet_tf.2.0.h5")) # Download the model via this link https://
 ↪github.com/OlafenwaMoses/ImageAI/releases/tag/1.0
prediction.loadModel()

predictions, probabilities = prediction.classifyImage(os.path.join("/home/
 ↪jupyter/resources/out/pngs/image-0002199.png"), result_count=10)
for eachPrediction, eachProbability in zip(predictions, probabilities):
    print(eachPrediction , " : " , eachProbability)
```

```
WARNING:tensorflow:AutoGraph could not transform <function
Model.make_predict_function.<locals>.predict_function at 0x7f257c099290> and
will run it as-is.
Please report this to the TensorFlow team. When filing the bug, set the
verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full
output.
Cause: list index out of range
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING: AutoGraph could not transform <function
Model.make_predict_function.<locals>.predict_function at 0x7f257c099290> and
will run it as-is.
Please report this to the TensorFlow team. When filing the bug, set the
verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full
output.
Cause: list index out of range
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
```

```
2021-10-24 19:28:57.475828: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR
Optimization Passes are enabled (registered 2)
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/dat
a/imagenet_class_index.json
40960/35363 [==================================] - 0s 0us/step
49152/35363 [=========================================] - 0s 0us/step
abaya  :  40.905800461769104
miniskirt  :  11.937235295772552
suit  :  10.489000380039215
kimono  :  4.382539540529251
crutch  :  3.9769671857357025
lab_coat  :  2.8504569083452225
trench_coat  :  2.386859618127346
sweatshirt  :  2.300059422850609
mosque  :  1.9872883334755898
pajama  :  1.881725899875164
```